Projet tuteuré

-

Semestre 2

# C:\Users\Sébastien\git\Projetcalculatrice\ProjetS2\logo.pngIUT Informatique Rodez - 1ère année -2014/2015-

# Sujet : Mini-Calculatrice / Mini-Tableur

# -En Java-

MIQUEL Jonathan, BAUBE Maxime, PERIES Mickaël, GRANIER Vincent, SANCHEZ Sébastien

Dossier-Projet

# Sommaire :

[A - Plan Projet](#_Toc420843303)

[I - Introduction](#_Toc420843304)

[II - Organisation du projet](#_Toc420843305)

[- Présentation du projet :](#_Toc420843306)

[- Cahier des charges initial :](#_Toc420843307)

[- Nos objectifs](#_Toc420843308)

[- Finalités](#_Toc420843309)

[- Chef de projet : choix et rôle](#_Toc420843310)

[- Autres rôles importants](#_Toc420843311)

[- Livrables](#_Toc420843312)

[- Cadre du projet](#_Toc420843313)

[- Exigences / Risques identifiés](#_Toc420843314)

[- Planification](#_Toc420843315)

[- Bilan de la gestion de projet](#_Toc420843316)

[B - Spécifications](#_Toc420843317)

[Cahier des charges](#_Toc420843318)

[I - Mini-calculatrice simple](#_Toc420843319)

[II – Mini-Tableur](#_Toc420843320)

[III - Menu](#_Toc420843321)

[Les cas d'utilisations :](#_Toc420843322)

[Cas numéro 1 :](#_Toc420843323)

[Cas numéro 2 :](#_Toc420843324)

[Cas numéro 3 :](#_Toc420843325)

[Cas numéro 4 :](#_Toc420843326)

[Cas numéro 5 :](#_Toc420843327)

[Cas numéro 6 :](#_Toc420843328)

[Cas numéro 7 :](#_Toc420843329)

[Cas numéro 8 :](#_Toc420843330)

[Cas numéro 9 :](#_Toc420843331)

[Cas numéro 10 :](#_Toc420843332)

[Cas numéro 11 :](#_Toc420843333)

[C - Conception](#_Toc420843334)

[I - Explications de la méthode](#_Toc420843335)

[II - Diagrammes des classes UML de l'application](#_Toc420843336)

[D - Codage](#_Toc420843337)

[E - Tests](#_Toc420843338)

[1) Programmes de test](#_Toc420843339)

[2) Résultats des programmes de test](#_Toc420843340)

[- TestCommandesCalculatrice](#_Toc420843341)

[- TestVariable :](#_Toc420843342)

[- TestUtilitaires](#_Toc420843343)

[- TestCommandes](#_Toc420843344)

[3) Scénarios de test](#_Toc420843345)

[I - Calculatrice](#_Toc420843346)

[II - Tableur](#_Toc420843347)

[F- Bilan](#_Toc420843348)

[1) Bilan commun](#_Toc420843349)

[2) Bilans personnels](#_Toc420843350)

[- Sébastien SANCHEZ](#_Toc420843351)

[- Jonathan MIQUEL](#_Toc420843352)

[- Maxime BAUBE](#_Toc420843353)

[- Vincent GRANIER](#_Toc420843354)

[- Mickaël PERIES](#_Toc420843355)

# A - Plan Projet

## I - Introduction

Nous allons réaliser un projet tuteuré dans le cadre de notre première année de DUT Informatique. Ce projet aura pour but de créer une application de Mini-Calculatrice / Mini-Tableur. Nous avons décidé de développer cette application en mode graphique, puisque, étant un groupe constitué de 5 personnes, nous pouvons répartir plus de tâches entre les membres du groupe, et donc développer une application plus complète.

Ce développement se fera à l'aide de plusieurs concepts et outils, qui permettront de réaliser la définition des besoins, la planification, la conception de l'application (COO), le développement en Java, la création d'IHM et la gestion de projet.

## II - Organisation du projet

### - Présentation du projet :

L'application que nous allons réaliser se doit d'être simple et efficace dans son utilisation, la calculatrice doit permettre des calculs simples, avec parenthèse et utilisant des cases mémoire. Ces dernières pourront être gérés via un gestionnaire de mémoire. Le tableur devra permettre les mêmes opérations (excepté avec les cases mémoire). Ces calculs seront stockés dans la cellule du tableur du choix de l'utilisateur. On pourra effectuer des modifications du tableur via des commandes qui permettront par exemple de copier les valeurs de cellules, ou de les remettre à zéro.

### - Cahier des charges initial :

### 

### - Nos objectifs

Nous souhaitons développer ce projet en interface graphique, afin de mieux comprendre le fonctionnement des bibliothèques swing et awt en java. Cela permettra un accès plus rapide et simple à l'application, que n'importe qui pourra utiliser facilement. Cela permettra également au groupe de s'exercer à la mise en application du cours d'IHM dispensé en DUT Informatique. De plus, nous pensons qu'un tableur en mode console aurait été peu lisible. Nous comptons réaliser le projet dans sa totalité, toutes les commandes seront donc fonctionnelles dans le livrable final.

### - Finalités

Notre projet ne présentera aucune finalité financière, puisqu'il n'est pas destiné à la vente. La finalité stratégique de celui-ci étant simplement de nous faire découvrir le développement d'une "vrai" application informatique, et par conséquent de nous expérimenté sur la gestion de projet informatique, la conception orientée objet, la programmation orientée objet et la conception d'IHM.

### - Chef de projet : choix et rôle

Le chef de projet qui a été désigné pour diriger le développement de cette application est Sébastien SANCHEZ. Etant à l'aise en programmation, mais également en communication, il semble capable de diriger ce projet correctement, en pouvant aider les autres à avancer mais également en définissant les spécifications qui aideront les programmeurs dans leurs tâches.

Au sein du groupe, le chef de projet devra permettre à tous d'exprimer correctement leurs capacités (programmation, conception, IHM, communication... ) en leur fournissant des spécifications claires, associées à des tâches adaptées à chacun. En effet, nous sommes tous étudiants, et nous devons donc veiller à ce que le projet permette à tout le monde d'apprendre et de s'exercer, peu importe ses capacités. De plus, il devra également s'atteler aux tâches de programmation et de communication. En effet, pour le cas spécifique de notre projet, le chef de projet n'aura que peu de travail à fournir. Le principal travail étant la définition des tâches et la spécification (cahier des charges) et l'organisation assez simple grâce à l'utilisation de Git (voir plus bas : gestionnaire de configuration), le chef de projet pourra aider au développement de l'application.

### - Autres rôles importants

- Le gestionnaire de configuration : Mickaël PERIES sera chargé de cette mission, il devra s'assurer que chacun puisse avoir accès au projet, chez lui ou à l'IUT. Pour cela, il devra apprendre à utiliser Git, GitHub et EGit, puis aider les autres membres du groupe à l'utiliser.

- Le secrétaire : Le secrétaire sera chargé de rédiger les documents nécessaires au cours du projet. Ce rôle a dans un premier temps été attribué à Maxime BAUBE, mais nous ferons tourner ce rôle au sein du groupe.

### - Livrables

A la fin du projet, nous devrons livrer ce dossier (plan projet, spécifications, conception, codage, tests, conclusion), ainsi qu'un CD-ROM contenant également le dossier, un installateur de l'application, le code, et le PowerPoint qui sera utilisé pour la soutenance orale.

### - Cadre du projet

Qualité : Le cahier des charges devra être respecté dans sa totalité, l'application testée et fonctionnelle, et le dossier correctement rédigé.

Coûts : Le projet ne représentera aucun coût puisque nous ne sommes pas rémunérés. L'application ne sera pas vendue donc le retour sur investissement sera nul également.

Délais : Le projet dans sa totalité devra être livré le 1er juin au plus tard. La planification qui suit montre comment nous nous organiserons pour atteindre ce but.

### - Exigences / Risques identifiés

Le principal risque lié aux exigences est le délai, qui devra absolument être respecté. Ainsi, la planification devra être respectée au mieux pour ne pas engendrer de retard, et pour éviter de ne pas pouvoir fournir une application incomplète.

### - Planification

Nous avons suivi la planification suivante, mise au point par le Chef de projet, grâce à Microsoft Projects. De plus, les jalons ont été les réunions avec la MOA/MOE, dont les comptes-rendus suivent. Nous avons choisi de suivre un cycle en V, permettant de mieux gérer chaque méthode lorsqu'elle est implémentée dans l'application, puisqu'elles seront testées directement.

### 

### - Bilan de la gestion de projet

La planification initiale ne s'est pas éloignée de la réalité. Nous avons réussi à suivre les délais sans prendre de retard. Cependant, les durées des tâches n'ont pas été exactement celles prévues, par exemple, la mise en place des calculs simples a été plus courte que prévue, tout comme la gestion de la mémoire, à l'inverse, l'implémentation des parenthèses et des calculs à plusieurs opérandes a été beaucoup plus longue que prévue, nécessitant 2 développeurs, pendant plus de temps. Ces changements se sont finalement équilibrés en fin de projet. La gestion de projet s'est, elle déroulée sans encombre, l'entente entre les différents membres du groupe a été très bonne même dans les moments de stress, et de prise de décision, qui se sont finalement faites assez facilement, en écoutant les avis de chacun.

# 

# B - Spécifications

## Cahier des charges

Nous avons adapté le cahier des charges fourni par la MOA afin qu'il corresponde mieux à nos attentes, notamment au niveau de l'IHM.

Nous allons développer une Mini-calculatrice et un Mini-tableur en Java. Le programme sera divisé en deux parties indépendantes : une partie calculatrice qui comportera la mini-calculatrice de base et le gestionnaire de mémoire, et une seconde partie qui sera le mini-tableur.

Le résultat sera une application graphique complète avec laquelle nous pourrons lancer au choix la calculatrice ou le tableur.

### I - Mini-calculatrice simple

#### a) Interface graphique - fonctionnement général

L'interface graphique de la partie calculatrice sera simple :

- L'utilisateur pourra entrer les instructions de calcul qu'il souhaitera exécuter dans un champ modifiable.

- L'utilisateur appuie sur la touche "Entrée" du clavier, ou bien sur un bouton "Calculer" situé à droite du champ où il entrera les calculs à effectuer pour exécuter les commandes qu'il entre.

- Lorsque le calcul est effectué, le champ modifiable est vidé, et la commande entrée est affichée dans une aire de texte non modifiable suivie du résultat, affiché à la ligne, précédé du symbole "=", ou d'une erreur si la commande est incorrecte.

#### b) Fonctionnement du champ d'exécution de commandes : mode calculatrice simple

##### 1) Calculs simples

- Une instruction de calcul simple sera constituée de 3 parties :

*nombre opérateur nombre*

- Le nombre pourra être un entier ou un réel, l'utilisateur pourra entrer un nombre négatif à l'aide du signe "-" (ex : -15) et un réel grâce au signe "." (ex : 5.1) ; les deux signes pourront être combinés.

- Les opérateurs seront :"+" pour l'addition ; "-" pour la soustraction ; "\*" pour la multiplication ; "/" pour la division.

- L'utilisateur pourra insérer autant d'espaces qu'il le souhaite entre les différents composants de l'instruction, mais l'instruction ne pourra pas dépasser 75 caractères.

- Il pourra également entrer autant de nombre qu'il souhaite, séparés par des opérateurs (ex : 1+2\*1+5/2)

##### 2) Calculs avec parenthèses

L'utilisateur pourra utiliser des parenthèses pour effectuer des calculs plus complexes.

- L'utilisateur devra obligatoirement fermer une parenthèse ouverte

- Les calculs dans les parenthèses seront effectués en premier

- Les lignes de commandes pourront avoir les formes suivantes avec les parenthèses :

(5 + 4) \* (3 + 1)

(5 \* (4 + 2))

(5 + 1)

- On autorisera également les calculs avec plusieurs opérateurs sans parenthèses (ex : 1+5\*4/2 )

- On autorisera autant de niveaux de parenthèses que l'on souhaite (mais la commande ne doit pas excéder 75 caractères).

##### 3) Calculs utilisant la mémoire

La calculatrice disposera de 26 mémoires nommées par les lettres majuscules de l'alphabet, de A à Z. L'utilisateur pourra utiliser ces mémoires dans ses calculs.

- Initialisation d'une mémoire à une valeur : syntaxe : 3 + 5 = A

- Le calcul sera tout de même effectué et le résultat affiché

- La mémoire pourra être utilisé dans les calculs comme un nombre "normal".

- Les zones mémoires pourront être consultées et éditées en mode gestionnaire de mémoire (voir plus bas).

Ex : 9 + 4 = A

= 13

A+2\*(1+3)

= 21

##### 4) Erreurs dans les calculs

- Si l'utilisateur n'entre pas un réel ou une lettre majuscule en tant qu'opérandes, on renverra une erreur. (ex : 1 + sd

- Si l'opérande est erronée ( pas +-\*/) ou que l'utilisateur en entre plusieurs à la suite (sans opérandes : ex : 1+\*1), on renverra une erreur.

- Si l'utilisateur entre un réel avec une virgule au lieu d'un point, on renverra une erreur.

- S'il utilise un point sans chiffres avant ou après (ex : 1. / .1), on renverra une erreur.

- Si l'utilisateur commence son calcul par un opérateur (ou le termine) ex : \* 2 ; 2 /

- Si l'utilisateur utilise dans un calcul une variable non initialisée

- Si l'utilisateur entre une variable à affecter qui n'est pas une lettre majuscule

- S'il n'utilise pas le signe "=" pour affecter un résultat de calcul à une variable

- S'il ne referme pas une parenthèse

- S'il entoure une affectation de parenthèse : ex : (1+2=A)

- S'il fait toute autre faute de syntaxe de calcul

- S'il entre une fonction du gestionnaire de mémoire sans être en mode MEM

#### c) Fonctionnement du champ d'exécution de commandes : mode mémoire

Lorsque l'utilisateur entrera l'expression "MEM" dans le champ d'exécution de commandes, la calculatrice passera en mode gestion de la mémoire.

Ce mode permettra de faciliter les opérations faisant intervenir la mémoire de la calculatrice.

- La commande "QUIT" permettra de revenir au mode "normal" de la calculatrice.

- L'utilisateur pourra utiliser la commande "AIDE" précédemment définie, ou le bouton "AIDE" pour consulter les différentes commandes disponibles.

- Les commandes permettant de gérer la mémoire sont composées de 3 parties :

--> Le nom de la commande

--> La zone de la mémoire concernée (cette partie peut être vide)

--> Une valeur numérique (pour certaines commandes)

Ex : MUL A 5

- La zone de mémoire concernée peut être un intervalle (plusieurs mémoires éditées en même temps. Pour cela, l'utilisateur devra préciser la première et la dernière zone mémoire concernée, et entrer ".." entre celles-ci (ex : A..F)

- Comme pour le mode normal, l'utilisateur pourra entrer autant d'espace qu'il le souhaite entre les composants de la commande, mais celle-ci ne pourra pas excéder 75 caractères.

Commandes disponibles :

- RAZ : remet à zéro les cases mémoires spécifiées ou toutes les cases mémoire et renvoie le message "OK" (ex : RAZ A..H)

- INCR : Ajoute 1 aux cases mémoires spécifiées, et renvoie le message "OK" (ex : INCR Y)

- SOM : effectue la somme des cases mémoires spécifiées, et renvoie le résultat (ex : SOM A..G)

- PROD : effectue le produit des cases mémoires spécifiées, et renvoie le résultat (ex : PROD A..B)

- MOY : effectue la moyenne des cases mémoires spécifiées, et renvoie le résultat (ex : MOY A..Z)

- SQRT : modifie la valeur des cases mémoires spécifiées pour leur affecter la racine carrée de leur valeur initiale, et renvoie "OK" (ex : SQRT A)

- CAR : modifie la valeur des cases mémoires spécifiées pour leur affecter le carré de leur valeur initiale, et renvoie "OK" (ex : SQRT G..K)

- INIT : Initialise la zone mémoire avec la valeur spécifiée, et renvoie "OK" (ex : INIT A 5)

- ADD : Ajoute la valeur spécifiée à la valeur des cases mémoires concernées et renvoie "OK" (ex : ADD G..L 42)

- MUL : Multiplie la valeur des cases mémoires concernées par la valeur spécifiée et renvoie "OK" (ex : MUL S 10)

- EXP : Elève la valeur de chacune des cases de la zone mémoire avec la puissance spécifiée par la valeur et renvoie "OK" (ex : EXP V 4)

#### d) Erreurs

- En mode MEM, s'il entre un calcul

- En mode MEM, s'il entre une commande non disponible

- S'il entre les plages de variables dans le désordre (ex : D..A au lieu de A..D)

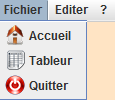
- Si, pour certains cas (ex : CAR) il utilise une ou plusieurs variables non initialisées

- Si les cases mémoires spécifiées ne sont pas des majuscules

- Si les plages ne sont pas séparées par ".." (ex : A,D ; A;D)

#### d) Menu

**Menu Fichier :**

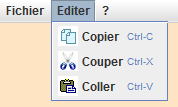


Accueil : permet de retourner au menu après avoir demandé de sauvegarder le tableur dans un fichier.

Tableur : permet de se déplacer directement vers le tableur.

Quitter : permet après confirmation par l’utilisateur de quitter complètement l’application.

**Menu Editer :**

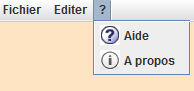


Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

**Menu Autre :**



A propos : Donne des informations sur les créateurs de l’application.

Aide : Affiche une nouvelle fenêtre contenant une aide approfondie pour aider l’utilisateur.

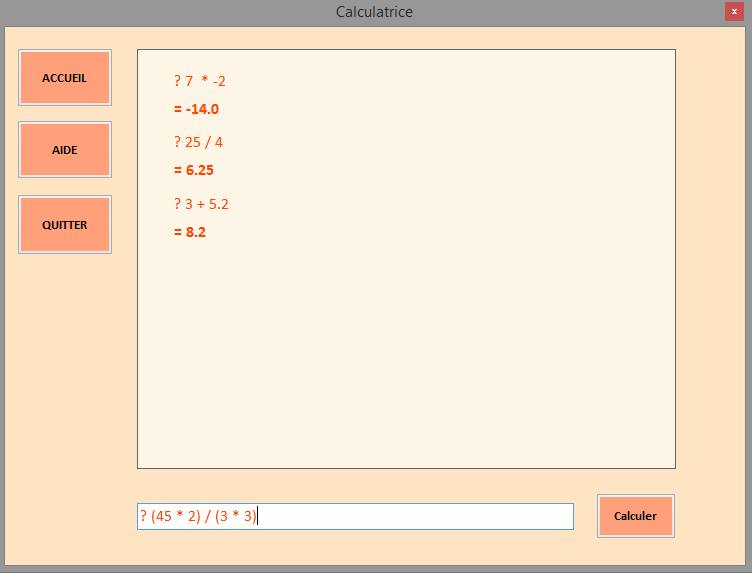
Menu pop-up : (s’affiche lors du clic droit dans le tableur)

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

#### e) Première maquette de l'interface de la calculatrice



### II – Mini-Tableur

#### a) Interface Graphique – fonctionnement général

L’utilisateur peut entrer des commandes dans la ligne de commande, ou directement sur une cellule, il devra donc entrer le nom de la case cible uniquement dans ligne de commande.

Il existe 3 principaux types d’instructions : Le simple affichage ou le texte entré sera simplement stocké dans la cellule destination, le calcul disponible en ajoutant le symbole ‘=’ et le dernier type d’instruction est l’entrée d’une commande.

La touche ‘Entrée’ permet de réaliser la même action que le bouton ‘Valider’ disponible.

Les deux options précédentes demanderont une confirmation, pour éviter les "fausses manipulations" des utilisateurs.

Un menu contiendra l’option "AIDE" permettra à l'utilisateur de mieux comprendre le fonctionnement de l'application. Il recensera toutes les commandes possibles et ce qu'elles font. On pourra également accéder à cette fonction en tapant la commande "AIDE".

#### b) Les types d’entrées possibles

##### 1) Via ligne de commande

L’utilisateur doit rajouter la (les) case(s) destination dans les commandes d’affichage et de calcul.

L’affichage simple permet d’afficher sans modifications le texte passé en paramètre.

Le calcul est réalisé en ajoutant un ‘=’ après la cellule cible, si une erreur se produit durant le calcul un simple affichage du calcul sera réalisé.

Les commandes ne nécessitent pas d’indiquer une cellule cible mais elles demandent des paramètres variables.

Commandes disponibles :

COPIER de type cellule/cellule cellule/plage plage/plage

RAZ cellule : supprime le contenu d’une cellule

RAZ plage : supprime le contenu d’une plage

AIDE : affiche l’aide

##### 2) Directement depuis une case

L’affichage simple ne nécessitera pas d’indiquer la cellule source.

Le calcul est réalisé en ajoutant le signe ‘=’.

#### c) Erreurs

En cas de tentative d’affichage d’une cellule inexistante.

En cas de commande inconnue ou avec des erreurs dans les paramètres.

En cas d’erreur de syntaxe sur un calcul (voir erreurs de la calculatrice partie calcul, les variables étant ici des noms de cellules

Si on utilise le contenu d'une case qui n'est pas une opérande valide pour un calcul

Si le nombre d'arguments est incorrect (ou le type) ex : COPVAL A1 A2 A3

Si on entre une plage dans le mauvais ordre (ex : D4 A1)

#### d) Menu

**Menu Fichier :**

Nouveau : permet d’ouvrir un nouveau tableur après avoir demandé de sauvegarder l’ancien dans un fichier.

Charger : permet d’ouvrir un fichier contenant un tableur après avoir demandé de sauvegarder l’ancien dans un fichier à l’endroit voulu.

Sauver : permet de sauvegarder le tableur dans un fichier et un endroit choisi par l’utilisateur.

Accueil : permet de retourner au menu après avoir demandé de sauvegarder le tableur dans un fichier.

Calculatrice : permet de se déplacer directement vers la calculatrice simple après avoir demandé de sauvegarder le tableur dans un fichier.

**Menu Editer :**

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

**Menu Autre :**

A propos : Donne des informations sur les créateurs de l’application.

Aide : Affiche une nouvelle fenêtre contenant une aide approfondie pour aider l’utilisateur.

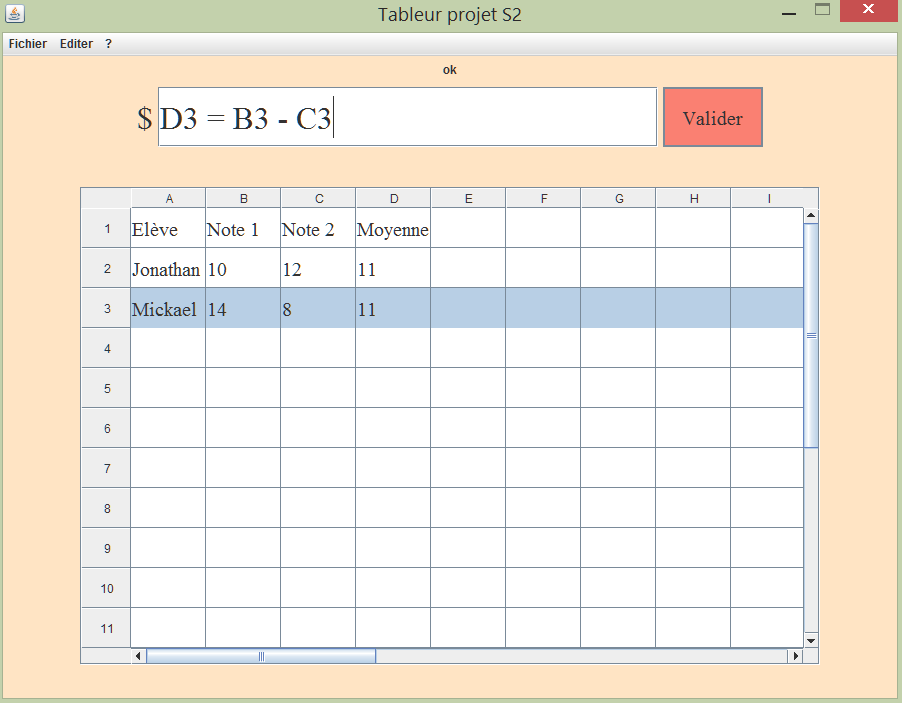
Menu pop-up : (s’affiche lors du clic droit dans le tableur)

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

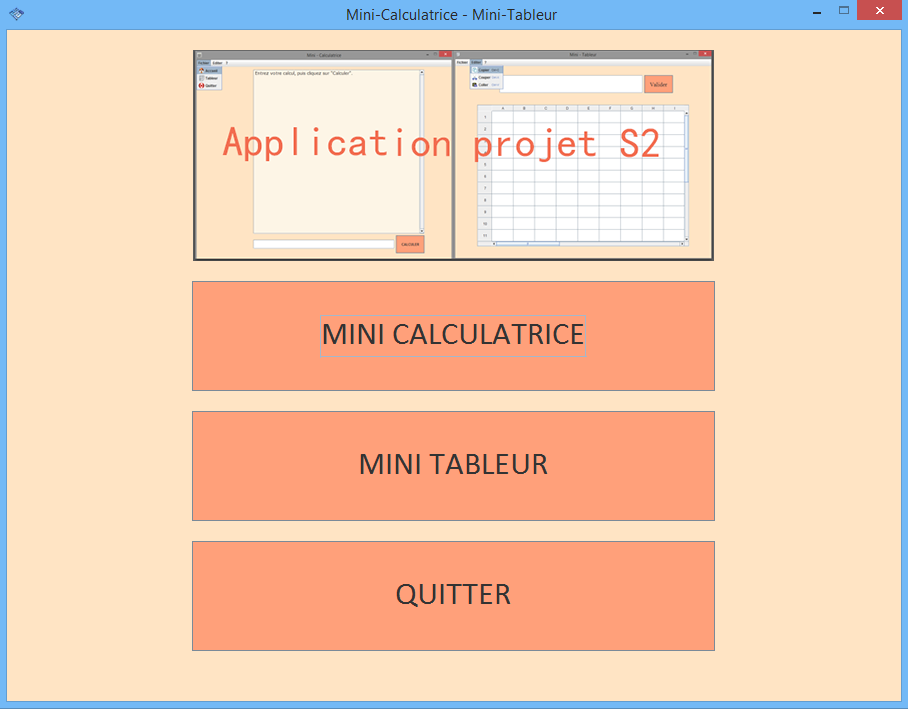
Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

#### e) Première maquette de l’interface graphique



### III - Menu



Lors de l’exécution de l’application, une fenêtre s’ouvre. Cette fenêtre est le menu de l’application. Ce menu nous propose trois actions :

MINI-CALCULATRICE : Permet d’accéder à la partie calculatrice de l’application

MINI-TABLEUR : Permet d’accéder directement au tableur sans avoir à passer par la commande TAB que nous aborderons par la suite

QUITTER : Permet de quitter l’application. Pour quitter cette application, deux offres possibilités s’offre à l’utilisateur : soit il passe par la croix rouge et un message de confirmation s’affiche alors, soit il est sur la fenêtre calculatrice et il choisit de fermer l’application en appuyant sur la touche entrée après avoir saisi une chaîne vide. La calculatrice lui demandera alors de confirmer son choix.

## Les cas d'utilisations :

Le diagramme UML qui suit est le diagramme des cas d'utilisations de l'application. Ils seront expliqués en détail dans les pages qui suivent.

### Cas numéro 1 :

#### Identification :

* Nom : Effectuer des calculs
* Objectif : détaille les étapes permettant à un utilisateur de réaliser des calculs
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Jonathan Miquel et Sébastien Sanchez
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, entre des calculs sur la console de la mini-calculatrice ou sur une case du mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Taper au clavier sur cette zone (la console/case du tableur)
3. La mini-calculatrice ainsi que le mini-tableur vérifie que le format du calcul soit correct
4. La mini-calculatrice et le mini-tableur renvoie le résultat

##### Enchaînement d’exceptions :

1. *Point 3 de la séquence nominale :* si le format est incorrect
   1. Affiche un message d’erreur
   2. Affiche un exemple de calcul

#### Rubrique optionnelle :

Les priorités des opérations sont prises en compte.

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

### Cas numéro 2 :

#### Identification :

* Nom : Effectuer des calculs simples
* Objectif : détaille les étapes permettant à un utilisateur de réaliser des calculs simples
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Sébastien Sanchez
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, entre des calculs « banals », c’est-à-dire sans parenthèses (du type : +,-,\*,/) sur la console de la mini-calculatrice ou sur une case du mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur le cham d’entrée des commandes
2. Taper au clavier sur cette zone (la console/case du tableur)
3. La mini-calculatrice ainsi que le mini-tableur vérifie que le format du calcul simple soit correct
4. La mini-calculatrice et le mini-tableur renvoie le résultat

##### Enchaînement d’exceptions :

1. *Point 3 de la séquence nominale :* si le format est incorrect
   1. Affiche un message d’erreur
   2. Affiche un exemple de calcul simple

#### Rubrique optionnelle :

Les priorités sur les opérations sont prises en compte.

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

### Cas numéro 3 :

#### Identification :

* Nom : Effectuer des calculs avec parenthèses
* Objectif : détaille les étapes permettant à un utilisateur de réaliser des calculs avec parenthèses
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Jonathan Miquel
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, entre des calculs avec parenthèses (du type : +,-,\*,/) sur la console de la mini-calculatrice ou sur une case du mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Taper au clavier sur cette zone (la console/case du tableur)
3. La mini-calculatrice ainsi que le mini-tableur vérifie que le format du calcul simple soit correct
4. La mini-calculatrice et le mini-tableur renvoie le résultat

##### Enchaînement d’exceptions :

1. *Point 3 de la séquence nominale :* si le format est incorrect
   1. Affiche un message d’erreur
   2. Affiche un exemple de calcul

#### Rubrique optionnelle :

Les priorités sur les opérations sont prises en compte (les priorités « classiques » et les parenthèses en plus).

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

### Cas numéro 4 :

#### Identification :

* Nom : Utiliser variable
* Objectif : détaille les étapes permettant à un utilisateur d’effectuer des calculs avec des variables.
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Sébastien Sanchez
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, entre des calculs simples ou avec parenthèses (du type : +,-,\*,/) sur la console de la mini-calculatrice ou sur une case du mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Taper au clavier sur ce champ (la console/case du tableur)
3. Initialiser la variable (pour cela passer en mode mémoire avec la commande MEM, puis initialiser avec INIT)
4. On peut ensuite utiliser cette variable pour les calculs suivants.
5. La mini-calculatrice ainsi que le mini-tableur vérifie que le format du calcul simple soit correct
6. La mini-calculatrice et le mini-tableur renvoie le résultat du calcul
7. On peut connaître la valeur de la variable avec la commande VOIR (il faut être en mode MEM)

##### Enchaînement d’exceptions :

1. *Point 3 de la séquence nominale :* si le format est incorrect
   1. Affiche un message d’erreur
   2. Affiche un exemple de calcul
2. *Point 4 de la séquence nominale :* si la variable n’est pas initialisée
3. Affiche un message d’erreur
4. Affiche un exemple d’utilisation

#### Rubrique optionnelle :

Les priorités sur les opérations sont prises en compte (les priorités « classiques » et les parenthèses en plus).

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

Les noms de variables à affecter sont : de A à Z(les minuscules ne sont pas prises en compte).

On peut utiliser plusieurs variables sur des calculs.

### Cas numéro 5 :

#### Identification :

* Nom : Gérer la mémoire
* Objectif : détaille les étapes permettant à un utilisateur de gérer la mémoire de l’application
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Sébastien Sanchez
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, décide de gérer la mémoire de la calculatrice à travers des commandes a entré sur la console de la mini-calculatrice ou sur une case du mini-tableur. Cette gestion peut se traduire par des commandes relatives à des fonctions mathématiques, l’affichage de l’aide, des copier/coller …etc.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Taper au clavier sur ce champ
3. Pour entrer dans le mode mémoire écrire la commande MEM
4. Ces commandes gèrent les variables déjà initialisées pour des calculs par exemple.

##### Enchaînement d’exceptions :

1. *Point 3 de la séquence nominale :* si l’utilisateur n’est pas dans le mode mémoire
2. Affiche un message d’erreur
3. Affiche un exemple d’utilisation
4. *Point 4 de la séquence nominale :* si l’utilisateur n’entre pas une commande mémoire valide
5. Affiche un message d’erreur adapté à la commande si l’utilisateur entre une commande proche de la vraie commande
6. Affiche un exemple d’utilisation adapté à la commande entré par l’user
7. Affiche un message d’erreur automatique si la commande entrée par l’utilisateur est totalement fausse
8. Affiche un exemple d’utilisation automatique si la commande entrée par l’utilisateur est totalement fausse

#### Rubrique optionnelle :

Les priorités sur les opérations sont prises en compte (les priorités « classiques » et les parenthèses en plus).

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

Les commandes doivent être en majuscules. Pour voir les commandes l’utilisateur doit se diriger vers l’aide.

### Cas numéro 6 :

#### Identification :

* Nom : Copier/Coller/Couper
* Objectif : détaille les étapes permettant à un utilisateur de copier,couper et coller les cases du mini-tableur.
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Mickaël Peries
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, réalise un clic droit ou clic sur l’option « Editer » de la barre d’outils.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Sélectionner l’opération, la variable, la valeur ou l’opérateur à copier/couper/coller
3. Réaliser un clic droit
4. Choisir les différentes options : copier,couper ou coller
5. L’autre option réalise les 2 premières étapes sauf qu’au lieu de réaliser un clic droit, l’utilisateur clic sur l’option »Editer » de la barre d’outils et choisit les différentes options

#### Rubrique optionnelle :

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

Ce cas d’utilisation peut être utilisé uniquement sur le champ d’entrée des commandes et non pas sur les cases pour le mini-tableur.

### Cas numéro 7 :

#### Identification :

* Nom : Entrer des formules
* Objectif : détaille les étapes permettant à un utilisateur d’entrer des formules sur le tableur
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Jonathan Miquel
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, peut entrer des formules sur le mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur le champ d’entrée des commandes
2. Définir la case dans laquelle le résultat va apparaître
3. Taper sur le clavier les commandes
4. Le mini-tableur vérifie si le format des formules est correct
5. Il affiche ensuite le résultat dans la case définie par l’utilisateur

##### Enchaînement d’exceptions :

1. *Point 2 de la séquence nominale :* si l’utilisateur ne définit pas la case qui accueillera le résultat de la formule
2. Message d’erreur signalant une erreur de syntaxe
3. *Point 3 de la séquence nominale :* si l’utilisateur ne respecte pas la syntaxe des formules
4. Message d’erreur

#### Rubrique optionnelle :

Les priorités sur les opérations sont prises en compte (les priorités « classiques » et les parenthèses en plus).

Le champ d’entrée des commandes/calculs ne peut contenir que 75 caractères.

Les commandes doivent être en majuscules. Pour voir les commandes l’utilisateur doit se diriger vers l’aide.

Les formules ne marcheront uniquement sur le champ d’entrée des commandes, et non pas sur les cases du tableur.

### Cas numéro 8 :

#### Identification :

* Nom : Sauvegarder, charger ou créer un nouveau fichier du tableur
* Objectif : détaille les étapes permettant à un utilisateur de sauvegarder, charger ou créer un nouveau fichier
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Mickaël Peries
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, veut sauvegarder, charger ou créer un nouveau fichier dans le mini-tableur.

##### Enchaînement nominal :

1. Cliquer sur l’option « fichier » de la barre d’outils
2. Pour créer un nouveau fichier cliquer sur « nouveau »
3. Pour sauvegarder cliquer sur « sauvegarder », on peut choisir l’emplacement du fichier et son nom
4. Pour charger un fichier déjà créer et sauvegarder il faut cliquer sur l’option « charger ».
5. Le mini-tableur affiche un message de confirmation lorsque que l’une de ces trois opérations est parfaitement réalisée.

##### Enchaînement d’exceptions :

1. *Point 5 de la séquence nominale* : si le fichier ne peut pas réaliser l’une de ces trois étapes pour cause de corruption de fichier
2. Affiche un message d’erreur

### Cas numéro 9 :

#### Identification :

* Nom : Accéder à une aide
* Objectif : détaille les étapes permettant à un utilisateur d’accéder à une aide.
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Vincent Granier et Maxime Baubé
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, souhaite accéder à une aide.

##### Enchaînement nominal :

1. Cliquer sur l’option « ? » de la barre d’outils
2. Cliquer sur « aide »

### Cas numéro 10 :

#### Identification :

* Nom : Naviguer entre Mini-Calculatrice et Mini-Tableur
* Objectif : détaille les étapes permettant à un utilisateur de passer de la Mini-Calculatrice au Mini-Tableur et inversement.
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Vincent Granier et Maxime Baubé
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur, après avoir ouvert son application, peut de la Mini-Calculatrice arriver au Mini-Tableur et inversement.

##### Enchaînement nominal :

1. Cliquer sur l’option « fichier » de la barre d’outils
2. Cliquer sur « Tableur » si vous êtes actuellement sur la Mini-Calculatrice ou cliquer sur « Calculatrice » si vous êtes sur le Mini-Tableur

#### Rubrique optionnelle :

Ce cas d’utilisation permet de rendre l’interface humain-machine plus rapide et efficace.

### Cas numéro 11 :

#### Identification :

* Nom : Quitter
* Objectif : détaille les étapes permettant à un utilisateur de quitter l’application
* Acteur principal : Utilisateur de la Mini-Calculatrice et du Mini-tableur
* Date : 01/06/2015
* Responsables : Mickaël Peries
* Version : 1.0

#### Séquencement :

Le cas d’utilisation commence lorsque l’utilisateur décide de fermer l’application. Plusieurs possibilités s’offrent à l’utilisateur pour quitter l’application.

##### Enchaînement nominal :

1. Lorsque vous êtes sur le menu il suffit de cliquer sur le bouton « quitter »
2. Lorsque que vous êtes sur la partie Mini-Calculatrice, cliquer sur l’option « Fichier » puis sur « quitter ». Même chose pour la partie Mini-Tableur.
3. Dans tous les cas vous pouvez fermer l’application avec la petite crois rouge en haut à droit de la fenêtre.

#### Rubrique optionnelle :

Par soucis de qualité de notre interface humain-machine nous avons décidé de demander la confirmation des choix de l’utilisateur lorsque ceux-ci était important.

Dans ce cas d’utilisation, l’action « quitter » est une action importante.

# C - Conception

## I - Explications de la méthode

Afin de développer cette application dans les meilleures conditions possibles, nous avons choisi de diviser au maximum le travail à faire, afin de permettre à tout le monde de développer une partie du projet, sans gêner les autres. Ainsi, nous avons d'abord divisé la conception du tableur, de la calculatrice, et du menu en 3 parties distinctes. Nous y avons ensuite ajouté un package comportant uniquement le lanceur de l'application, afin de bien repérer cette fonction qui est "à part" du reste du projet, puisqu'elle ne fait que lancer l'application. Nous avons ensuite ajouté un package de tests, pour pouvoir séparer les tests de l'application en elle-même. Enfin, nous avons décidé de séparer la partie graphique de la partie "métier" en ajoutant un package utilitaire, comportant les classes de méthodes utiles à l'exécution de commandes mais qui ne sont pas modifiées en fonction de l'interface. Du côté de la calculatrice, les actions de l'utilisateur se limitant à l'exécution de la commande via le bouton calcul, et à la navigation via le menu en haut de l'écran, nous avons décidé de laisser l'action performée par ce bouton avec le reste des classes liées à l'IHM, et d'isoler la partie navigation dans un sous-package. Enfin, du côté du tableur, nous avons décidé de créer un sous package contenant toutes les actions possibles de l'utilisateur, car leur nombre est beaucoup plus conséquent. Nous avons donc obtenu 8 packages distincts :

iut.info1.projetS2.calculatrice : IHM de la calculatrice

iut.info1.projetS2.calculatrice.navigation : partie navigation de la calculatrice

iut.info1.projetS2.executeur : lanceur de la calculatrice

iut.info1.projetS2.menu : Le menu principal de l'application

iut.info1.projetS2.tableur : IHM du tableur, commandes liées à cette IHM

iut.info1.projetS2.tableur.action : actions évènementielles de la partie tableur

iut.info1.projetS2.tests : programmes de tests

iut.info1.projetS2.utilitaires : classes contenant les méthodes liées à la partie "métier"

## II - Diagrammes des classes UML de l'application

Le diagramme UML qui suit présente la conception de l'application dans son ensemble.

# D - Codage

Les pages qui suivent présentent le code mis en place pour faire fonctionner l'application.

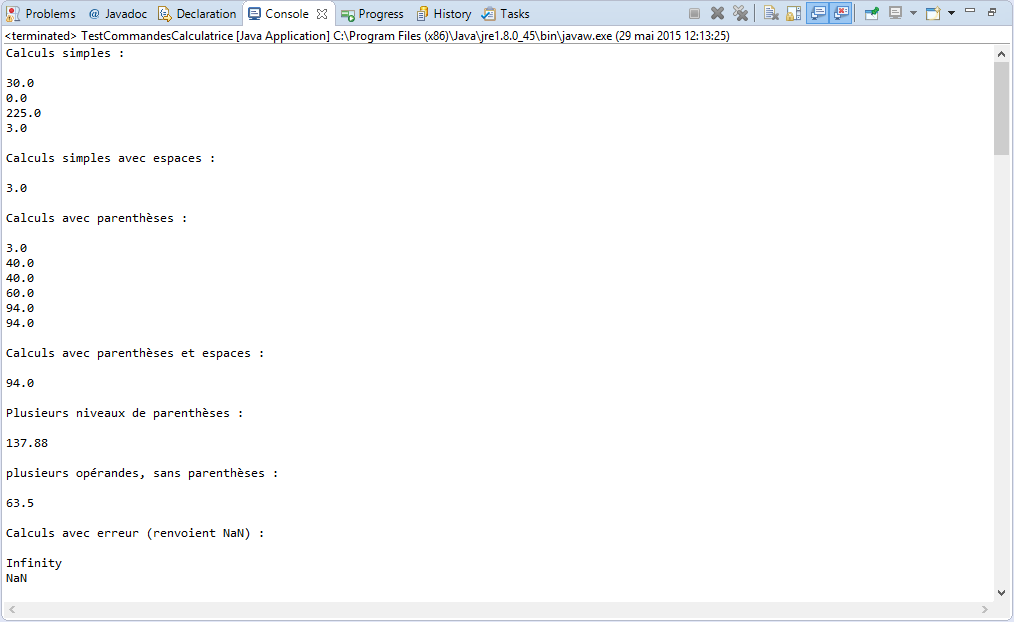
# E - Tests

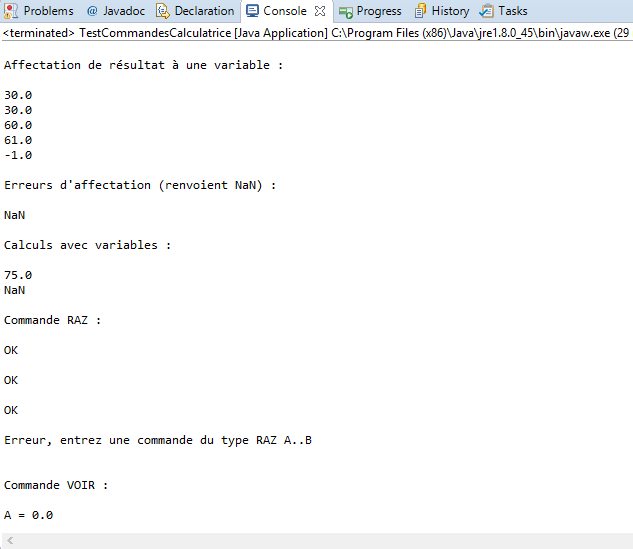
## 1) Programmes de test

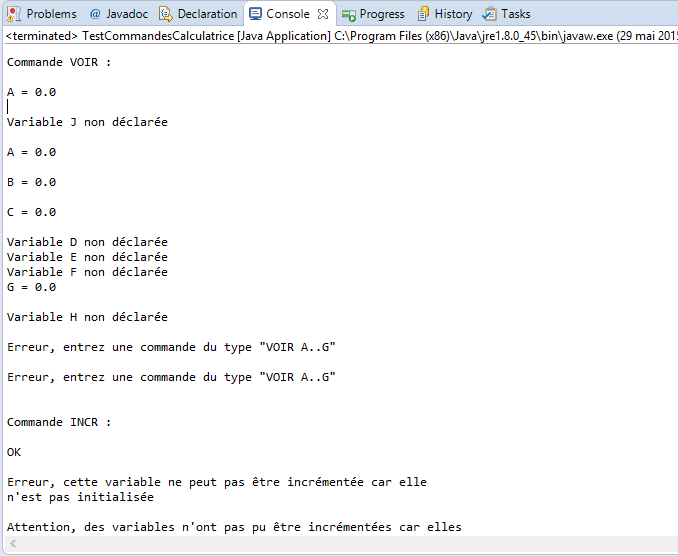
Voici les programmes de test développés pour tester les différentes parties de l'application

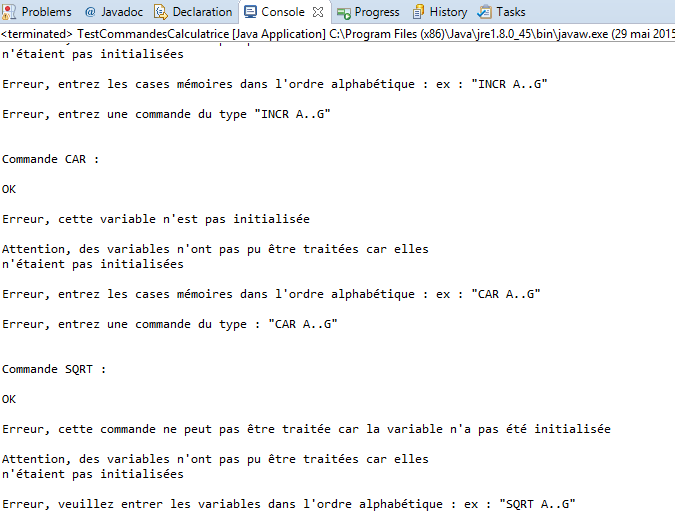
## 2) Résultats des programmes de test

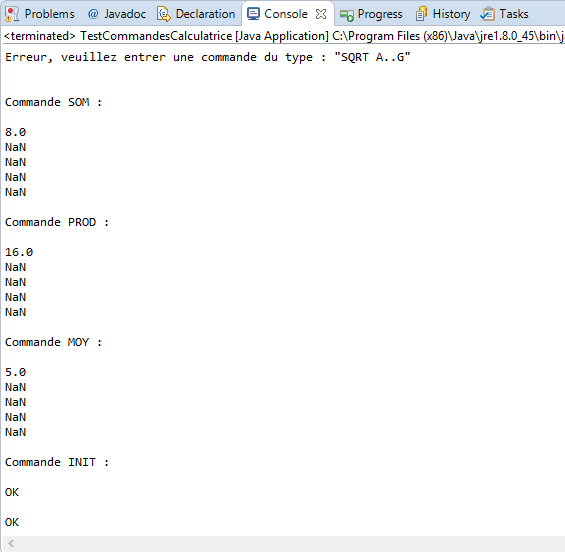
### - TestCommandesCalculatrice

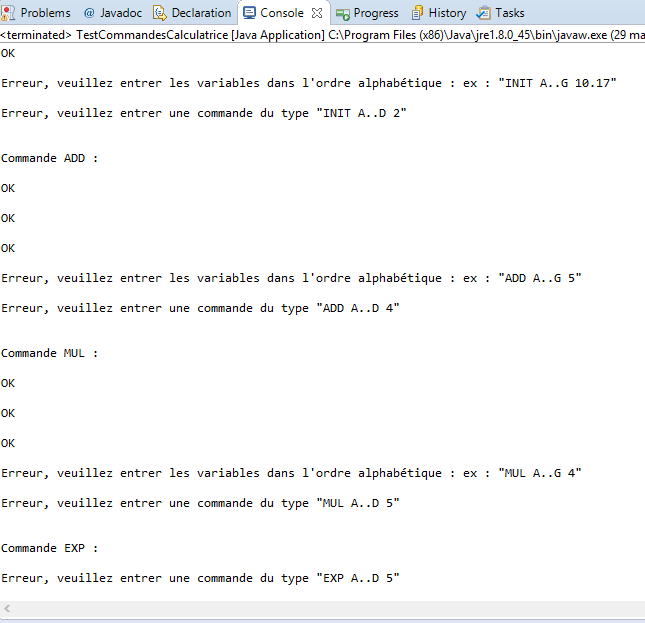


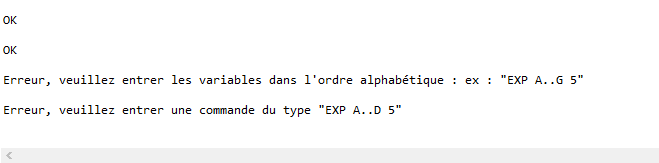








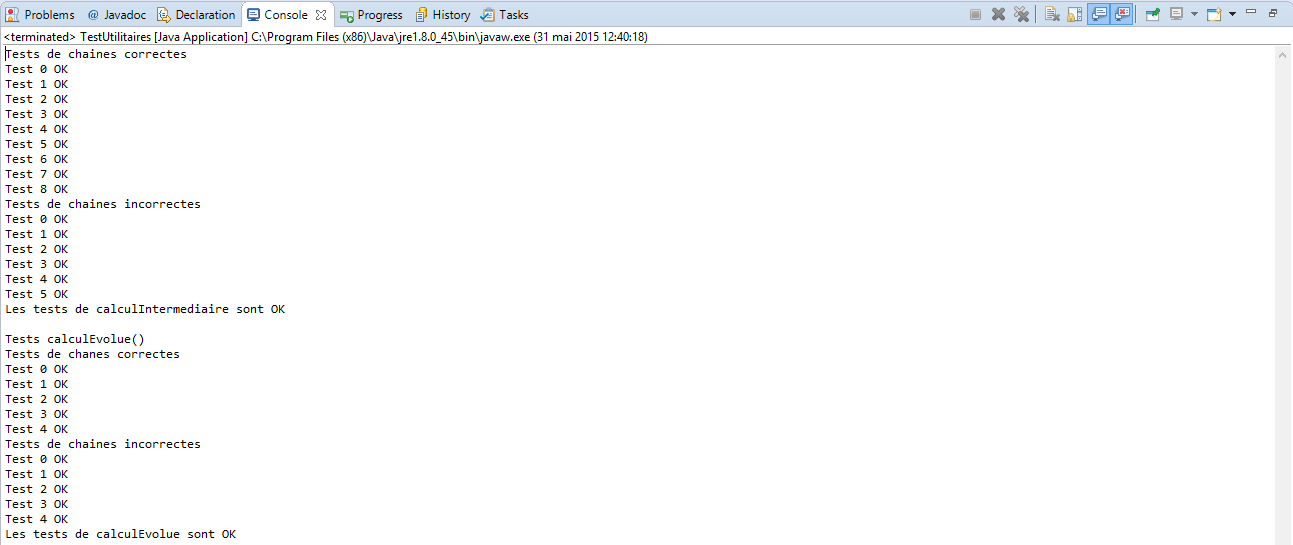




### - TestVariable :

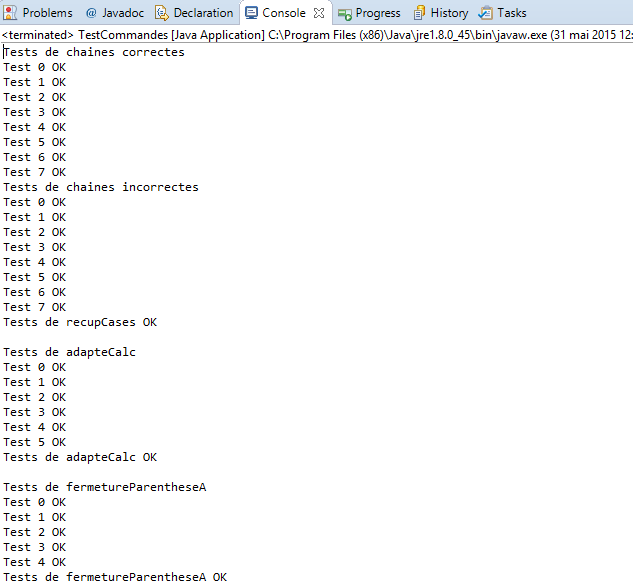
### 

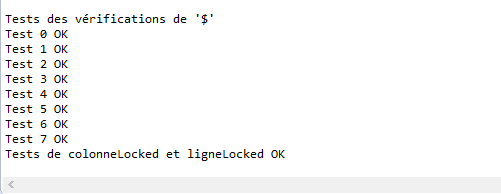
### - TestUtilitaires



### - TestCommandes

Ce test a été inséré dans le package contenant le fichier Commandes, sinon les tests auraient été impossibles à mettre en place.

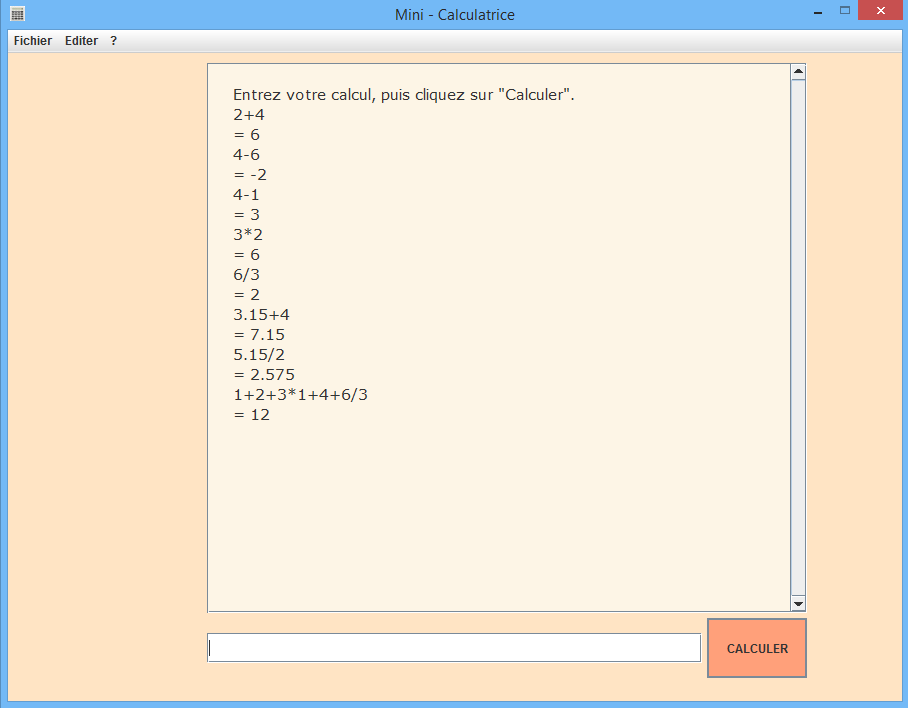


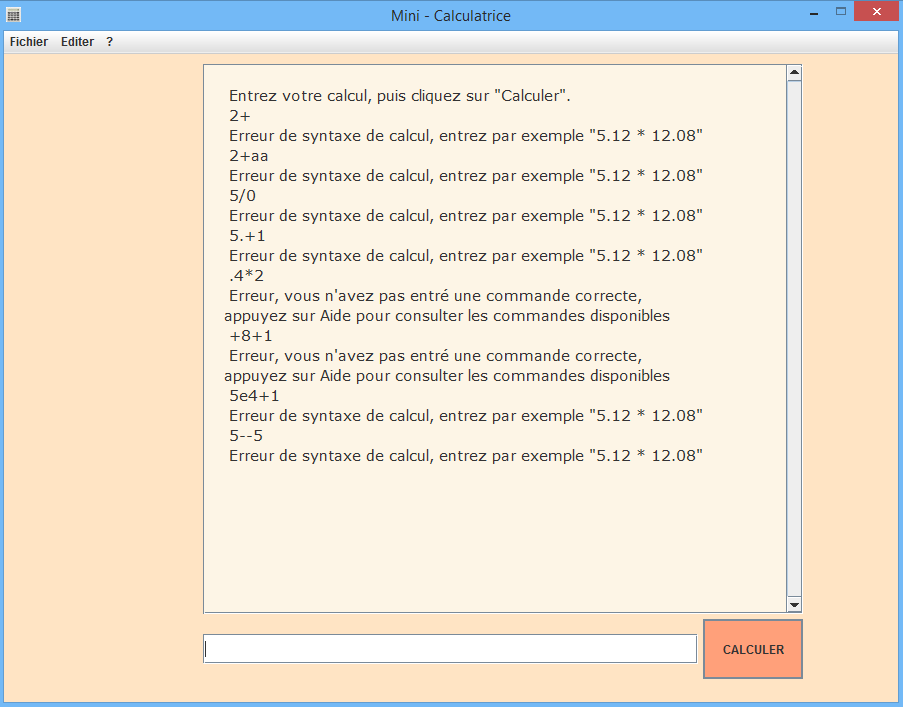


## 3) Scénarios de test

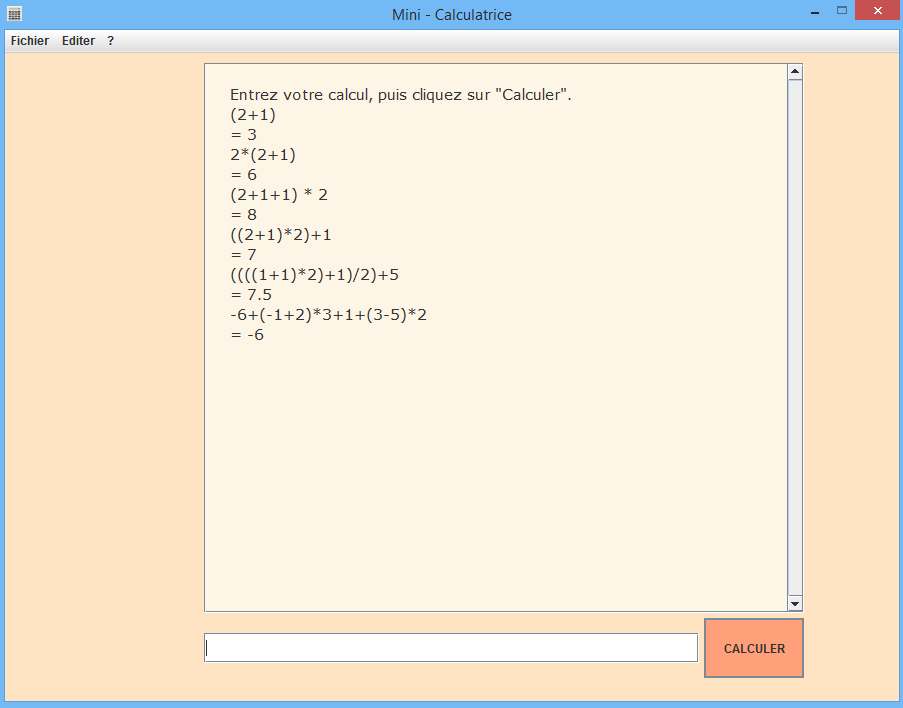
### I - Calculatrice

**- Calculs simples, avec 1 ou plusieurs opérandes**

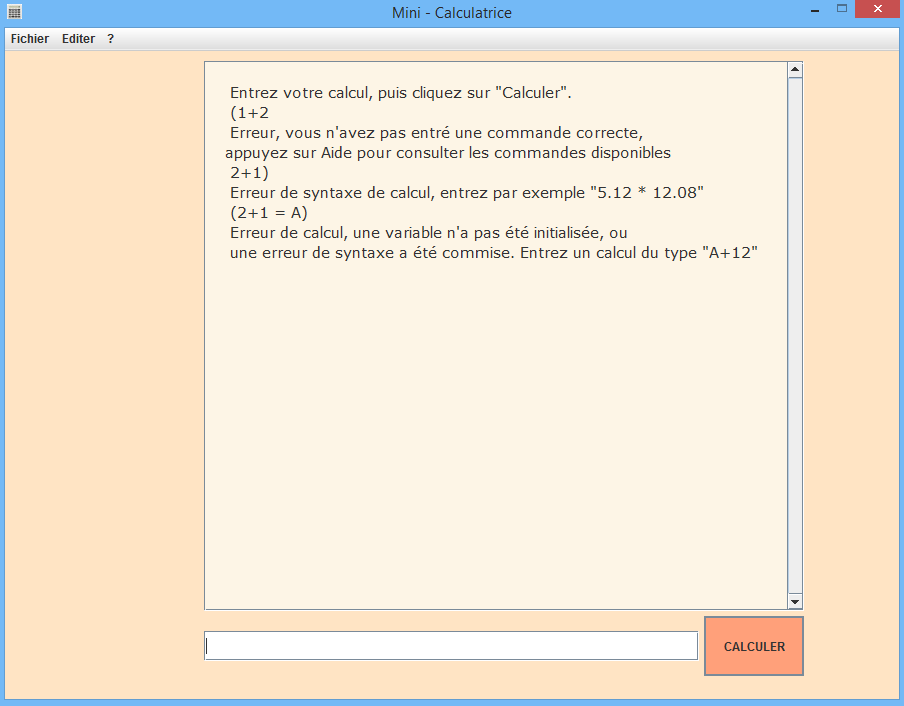
****

**Les Cas d’erreurs :  
  
**

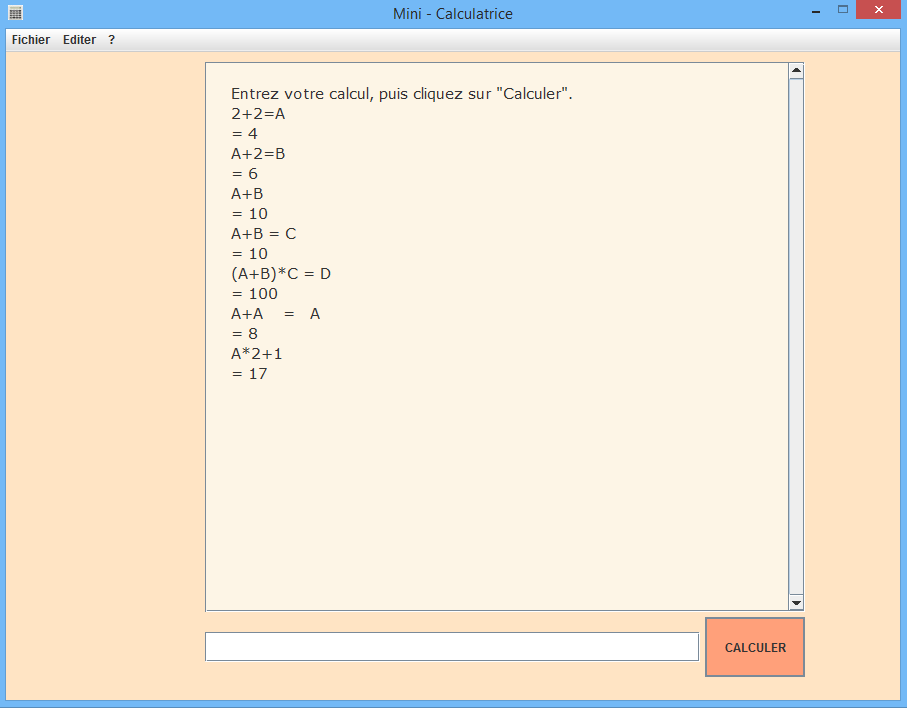
**- Calculs avec parenthèses**

****

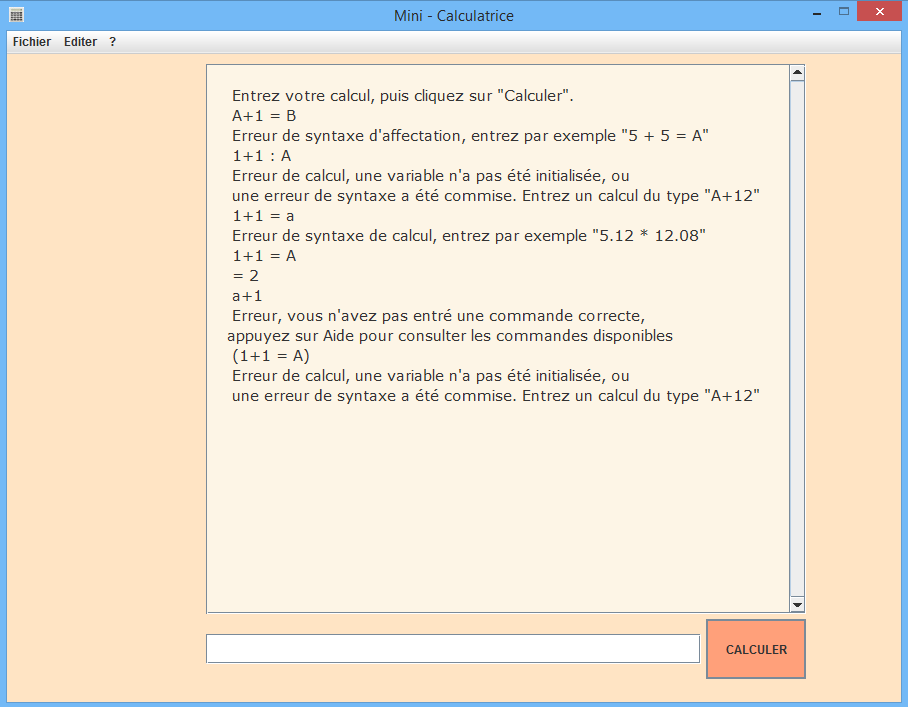
**- Erreurs :**

****

**Avec la mémoire :**

****

**- Erreurs :**

****

**- Mode MEM :**

Ici, les résultats et les commandes sont simplement copié-collés, le nombre d'impression écran aurait été énorme sinon, par rapport aux informations qu'elles auraient fournies.MEM

--> Mode mémoire actif.

QUIT

-->Mode mémoire inactif.

RAZ A

-->OK

RAZ A..D

-->OK

RAZ D..A

-->Erreur, entrez une commande du type RAZ A..B

RAZ g

-->Erreur, entrez une commande du type RAZ A..B

INIT A 5

-->OK

INIT B..D 4

-->OK

INIT A

-->Erreur, veuillez entrer une commande du type "INIT A..D 2"

INIT a 5

-->Erreur, veuillez entrer une commande du type "INIT A..D 2"

INIT D..A 5

-->Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "INIT A..G 10.17"

VOIR A

-->A = 4.0

VOIR B..D

-->B = 4.0

C = 4.0

D = 4.0

VOIR H

-->Variable H non déclarée

VOIR g

-->Erreur, entrez une commande du type "VOIR A..G"

VOIR D..A

-->Erreur, entrez une commande du type "VOIR A..G"

VOIR A-D

-->Erreur, entrez une commande du type "VOIR A..G"

INCR A

-->OK

INCR A..C

-->OK

INCR T

-->Erreur, cette variable ne peut pas être incrémentée car elle

n'est pas initialisée

INCR a

-->Erreur, entrez une commande du type "INCR A..G"

INCR C..A

-->Erreur, entrez les cases mémoires dans l'ordre alphabétique : ex : "INCR A..G"

INCR A..G

-->Attention, des variables n'ont pas pu être incrémentées car elles

n'étaient pas initialisées

AIDE

--> Ouvre la fenêtre d'aide

CAR A

--> OK

CAR A..C

--> OK

CAR T

--> Erreur, cette variable n'est pas initialisée

CAR A..R

--> Attention, des variables n'ont pas pu être traitées car elles

n'étaient pas initialisées

CAR D..A

--> Erreur, entrez les cases mémoires dans l'ordre alphabétique : ex : "CAR A..G"

CAR g

--> Erreur, entrez une commande du type : "CAR A..G"

SQRT A

--> OK

SQRT A..C

--> OK

SQRT T

--> Erreur, cette commande ne peut pas être traitée car la variable n'a pas été initialisée

SQRT A..R

--> Attention, des variables n'ont pas pu être traitées car elles

n'étaient pas initialisées

SQRT D..A

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "SQRT A..G"

SQRT g

--> Erreur, veuillez entrer une commande du type : "SQRT A..G"

SOM A..C

= 15

SOM A

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM C..A

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM A..H

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM a..c

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..C

= 5

MOY A

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY C..A

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..T

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..c

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..C

= 125

PROD A

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD C..A

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..T

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..c

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

ADD A 4

--> OK

ADD A..C 4

--> OK

ADD A

--> Erreur, veuillez entrer une commande du type "ADD A..D 4"

ADD a 4

--> Erreur, veuillez entrer une commande du type "ADD A..D 4"

ADD D..A 4

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "ADD A..G 5"

MUL A 2

--> OK

MUL A..C 2

--> OK

MUL A

--> Erreur, veuillez entrer une commande du type "MUL A..D 5"

MUL a 4

--> Erreur, veuillez entrer une commande du type "MUL A..D 5"

MUL D..A 2

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "MUL A..G 4"

EXP A 2

--> OK

EXP A..C 2

--> OK

EXP A

--> Erreur, veuillez entrer une commande du type "EXP A..D 5"

EXP a 2

--> Erreur, veuillez entrer une commande du type "EXP A..D 5"

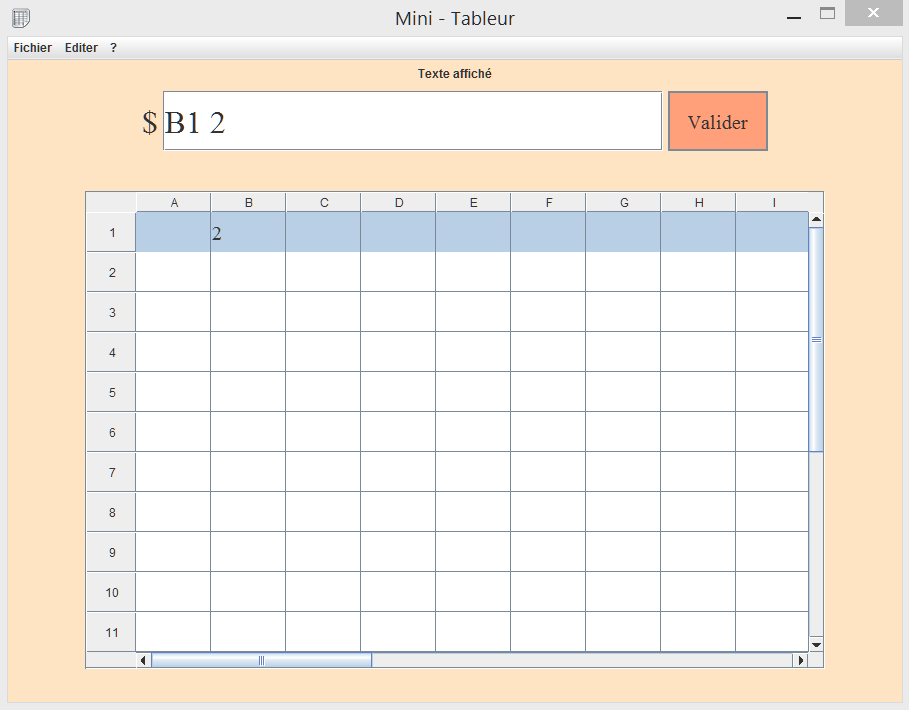
EXP D..A 2

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "EXP A..G 5"

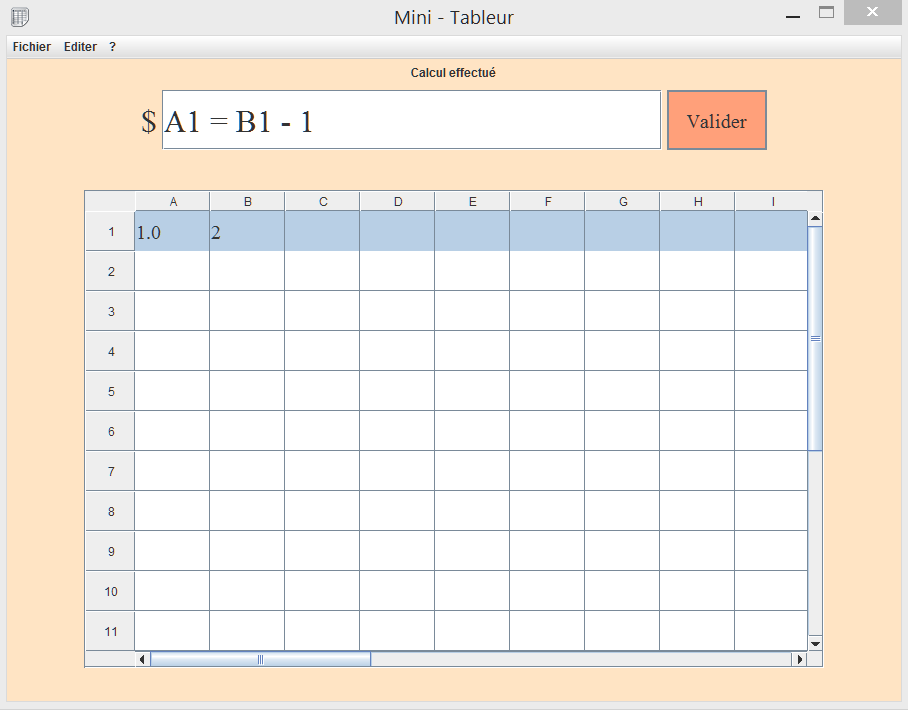
### II - Tableur

#### - Utilisation "normale"

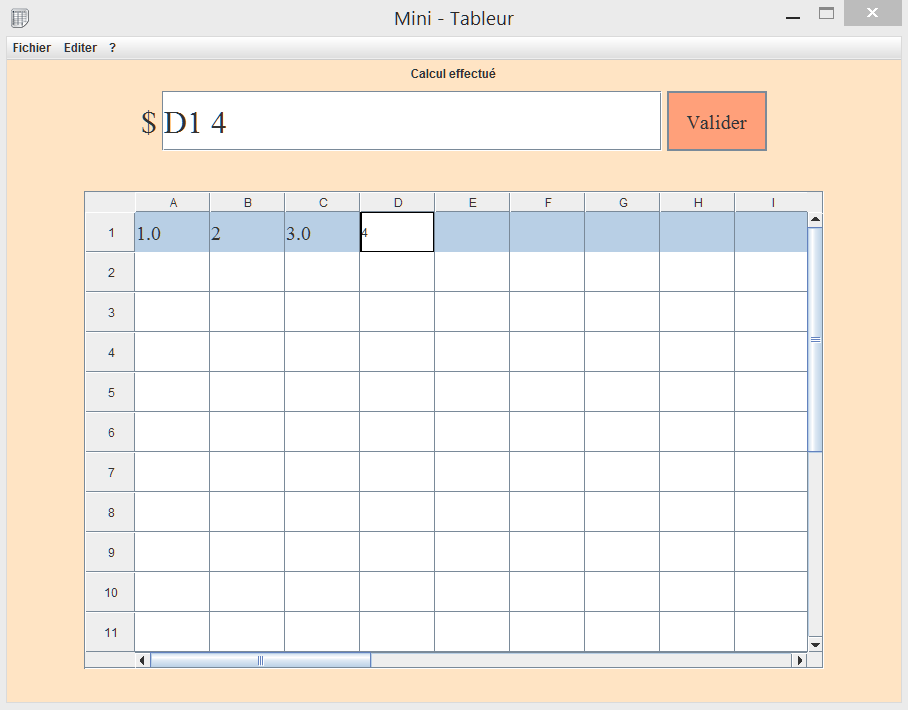
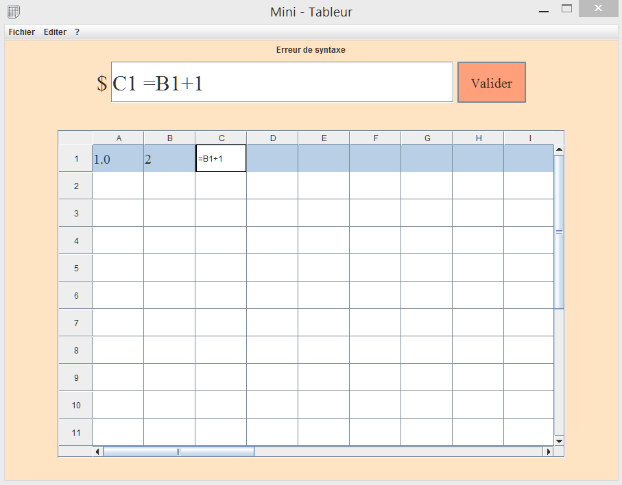
L’utilisateur veut se servir de la fonctionnalité la plus basique du tableur : L’affichage, il doit pour cela entrer dans la ‘ligne de commande’ la case destinatrice suivie d’au moins un espace suivi de son texte.



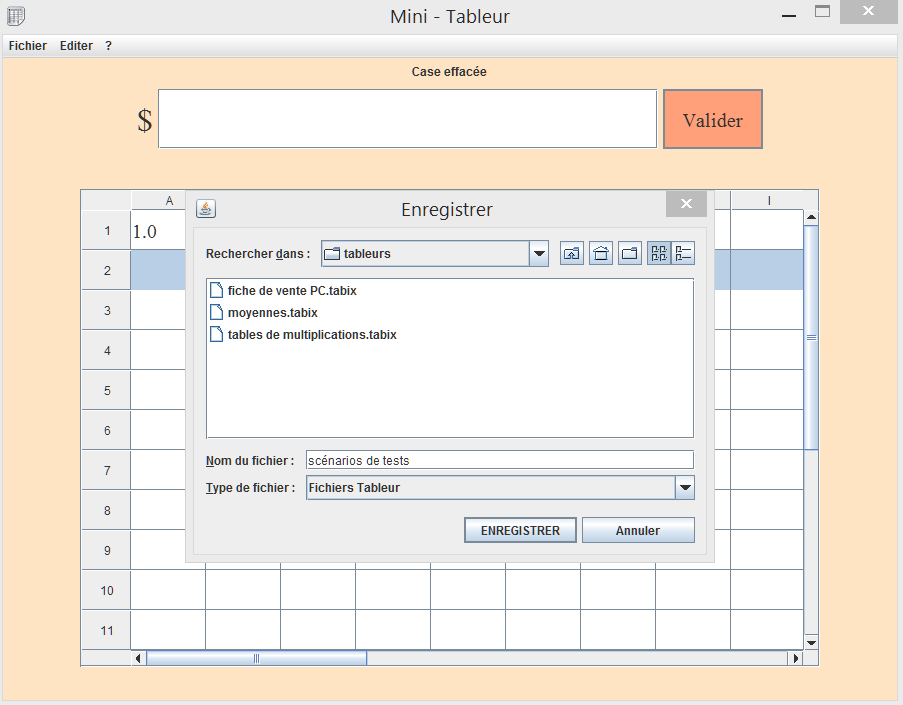
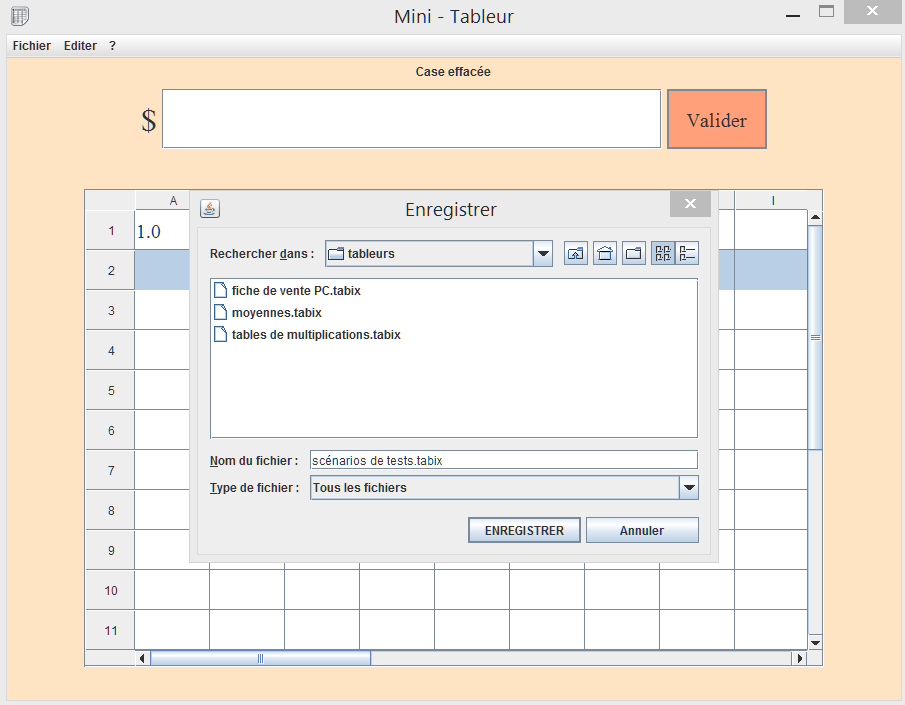
Une autre possibilité du tableur est la réalisation de calculs, pour ce faire l’utilisateur doit entrer dans la ligne de commande la case destinatrice suivie de 0 ou plusieurs espaces, suivie d’un caractère ‘=’ lui-même suivi du calcul souhaité, ce calcul peut contenir des cases qui seront remplacées par leur valeur avant le calcul.

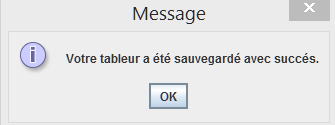


Les affichages simples, comme les calculs peuvent s’effectuer directement en écrivant le texte sur la case destinatrice il faudra donc simplement entrer le texte à afficher ou un ‘=’ suivi du calcul à effectuer.

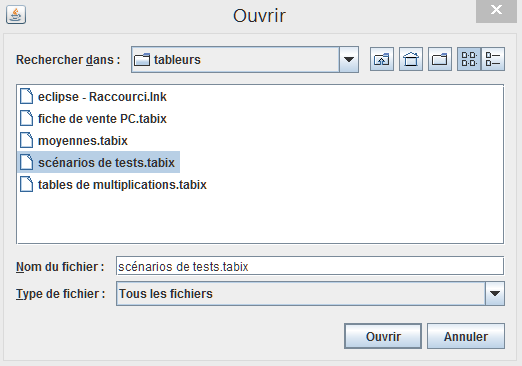


L’application propose également d’enregistrer l’état actuel du tableur, il suffit de cliquer sur fichier, un menu déroulant apparaît, et de choisir sauver. On pourra ainsi choisir le dossier d’enregistrement ainsi que le nom du fichier, l’extension qui lui sera associée sera « .tabix ».





Pour ouvrir fichier enregistré il faudra cliquer sur fichier puis sur ouvrir et une fenêtre de même type que celle de l’enregistrement permettra de choisir le fichier à ouvrir.

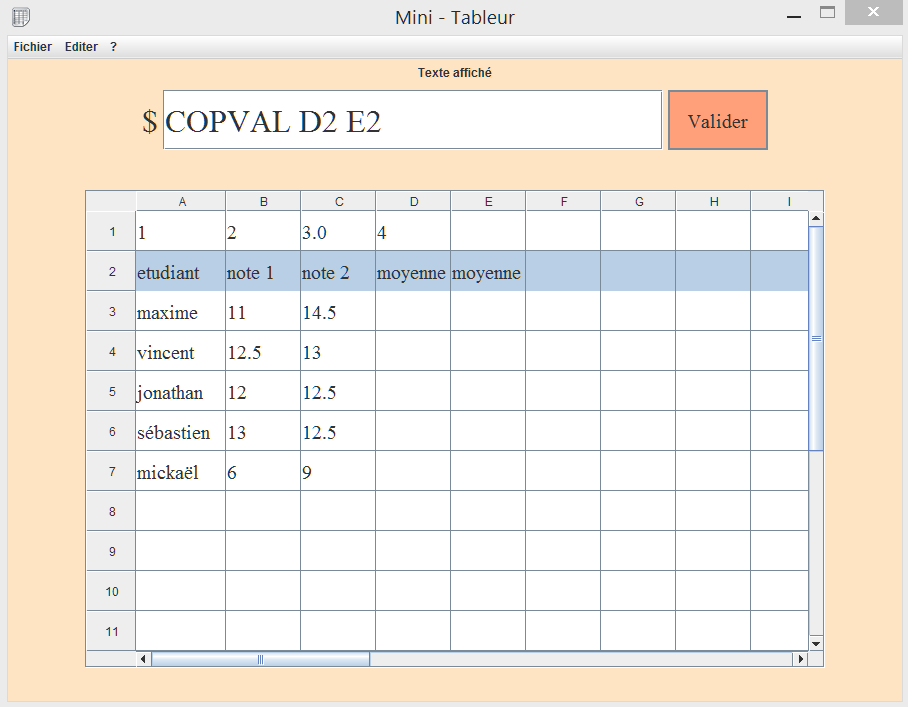


#### - Commandes

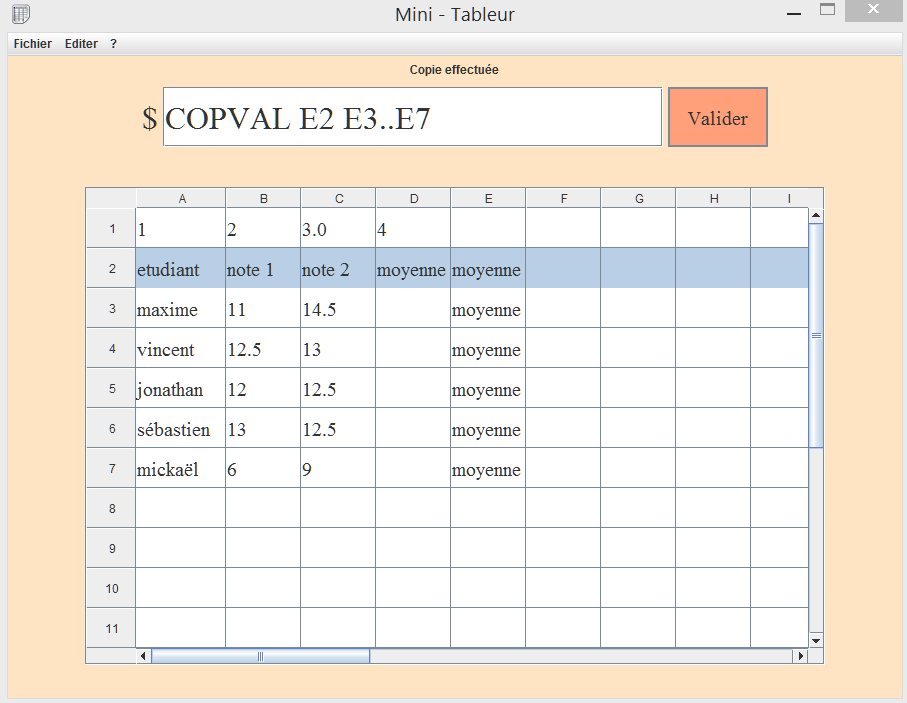
La dernière fonctionnalité du tableur est la possibilité d’entrer des commandes dans la ligne de commande, si elles sont entrées dans une case elle seront considérées comme du simple texte à afficher dans cette case.

##### COPVAL :

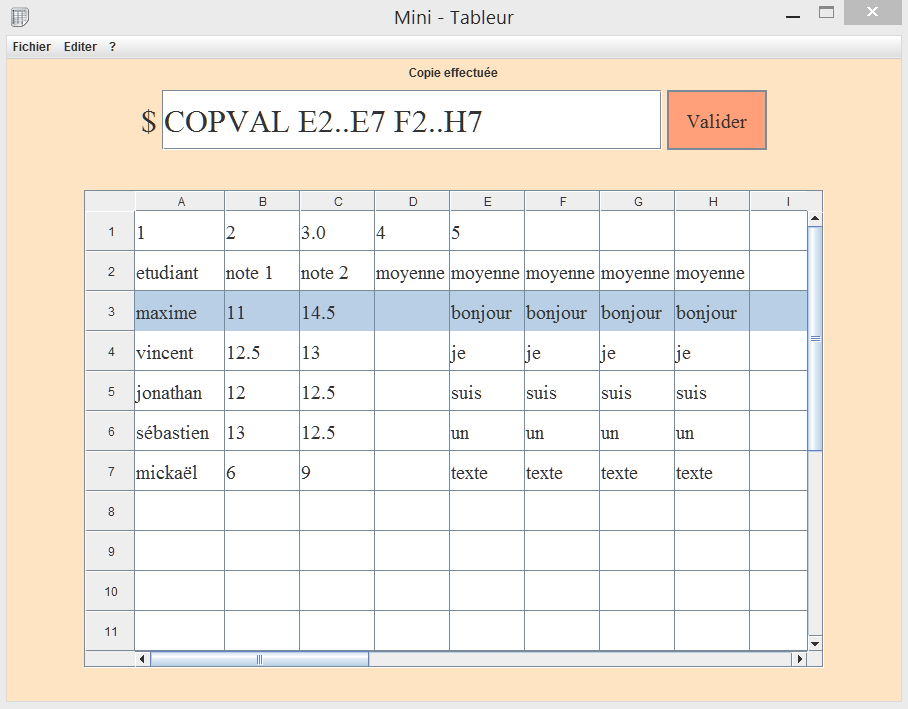
Elle permet de copier la valeur depuis -source- vers -destination- sans modifier celle-ci.

COPVAL CASE CASE : la source est une case tout comme la destination

COPVAL CASE PLAGE la source sera copiée dans chacune des cases de la plage entrée



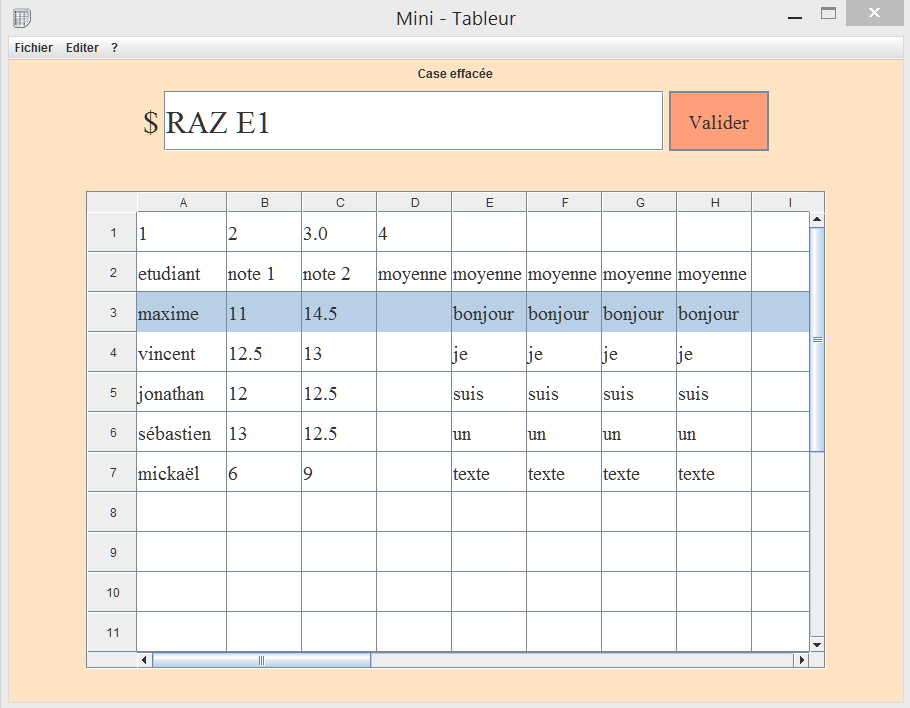
COPVAL PLAGE PLAGE : copier la plage source dans la plage destination



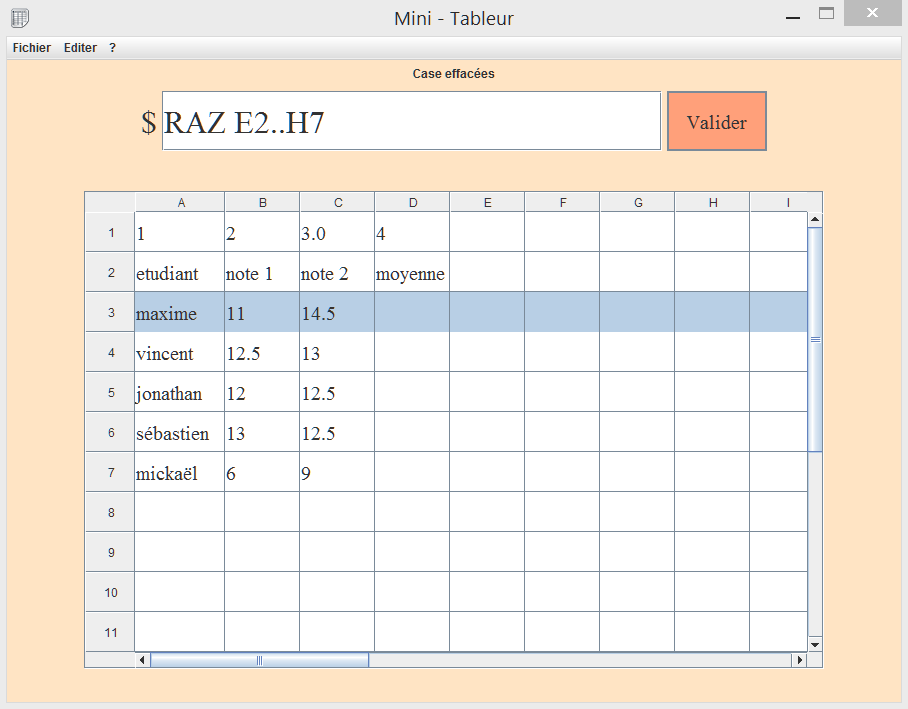
##### RAZ :

Elle permet de supprimer la case/plage spécifiée en paramètre.

RAZ CASE : Remet à sa valeur par défaut la case spécifiée



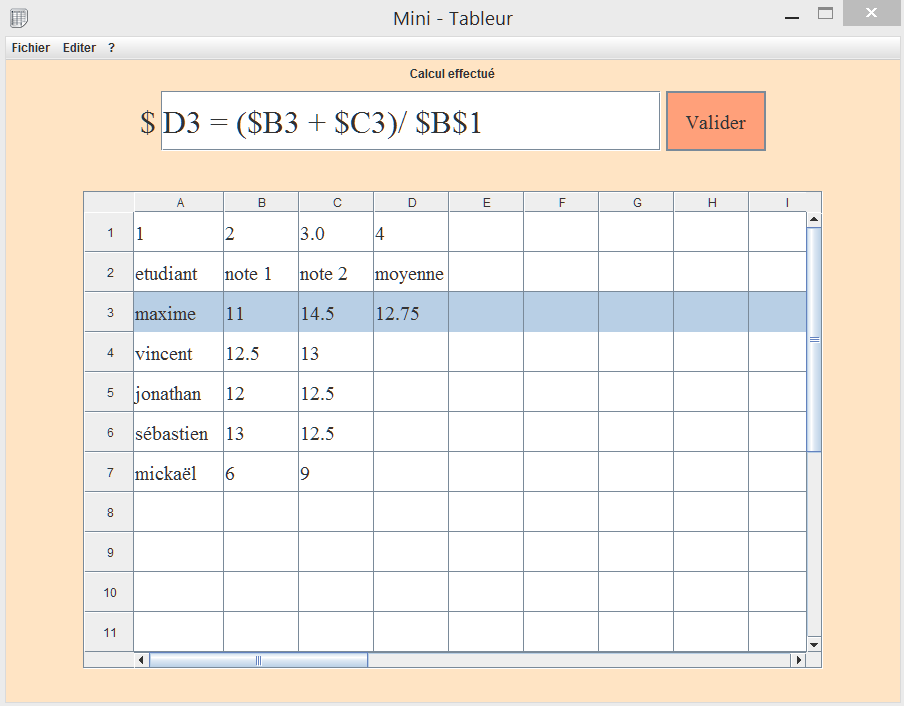
RAZ PLAGE : Remet à neuf toute la plage indiquée



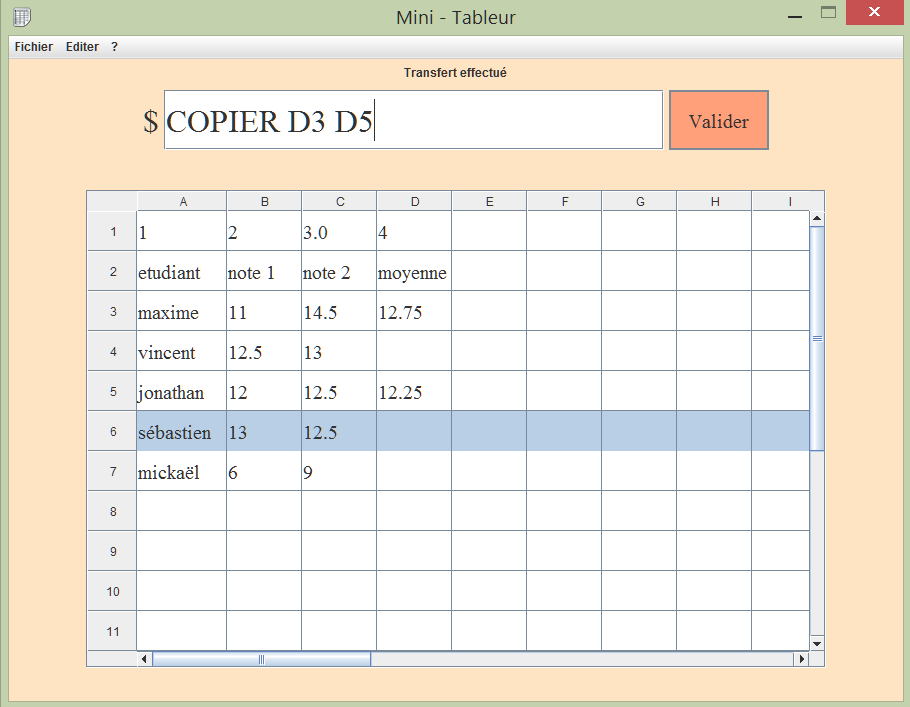
##### COPIER :

Tout comme COPVAL elle va copier la source vers la destination, mais en cas de calcul sur la cellule source impliquant d’autres cases, une adaptation sera effectuée l’utilisateur peut empêcher cette adaptation en rajoutant un ‘$’ devant la coordonnée à fixer.

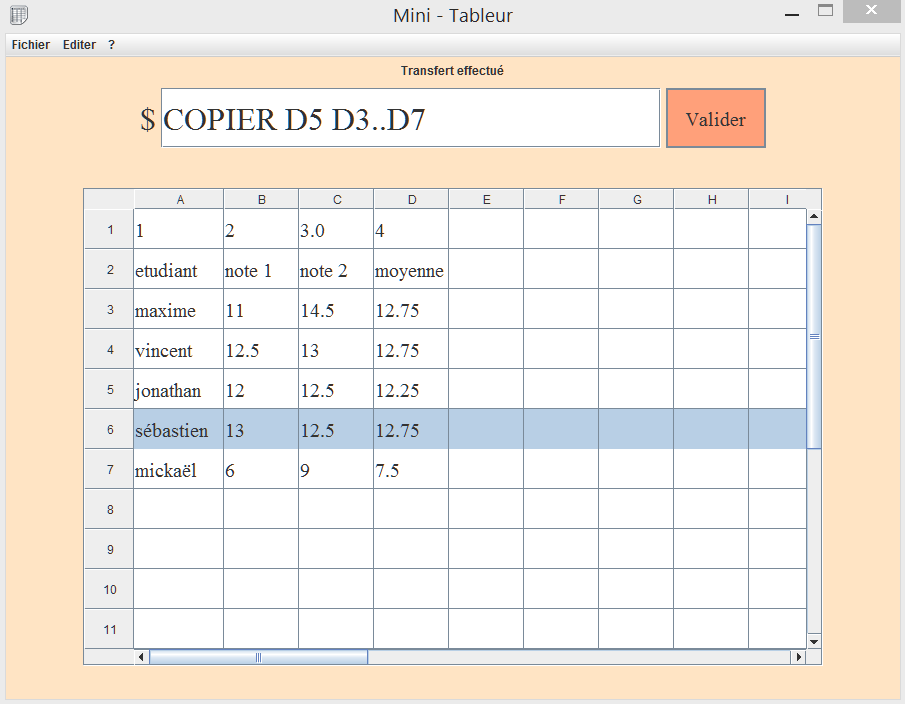
Calcul avec case :



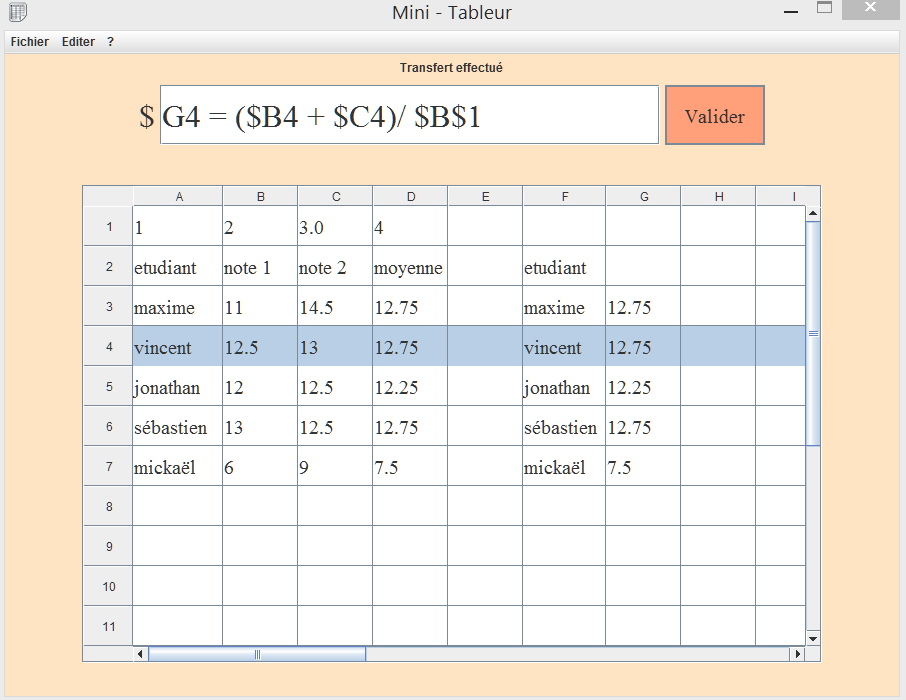
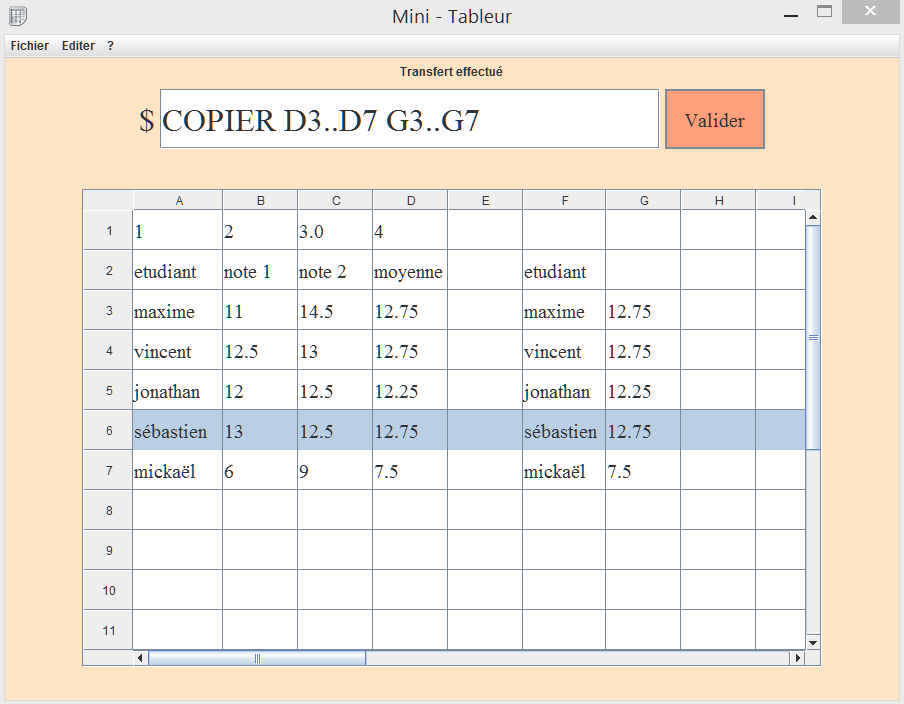
COPIER CASE CASE : copie simplement la case source vers la case destination



COPIER CASE PLAGE copie la case source dans chacunes des cases destination



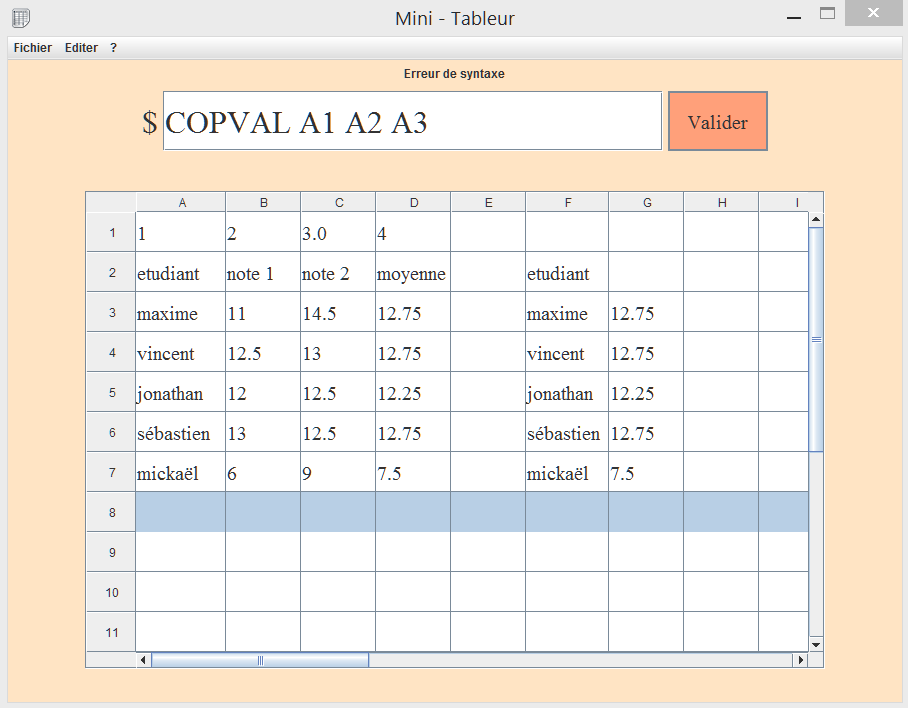
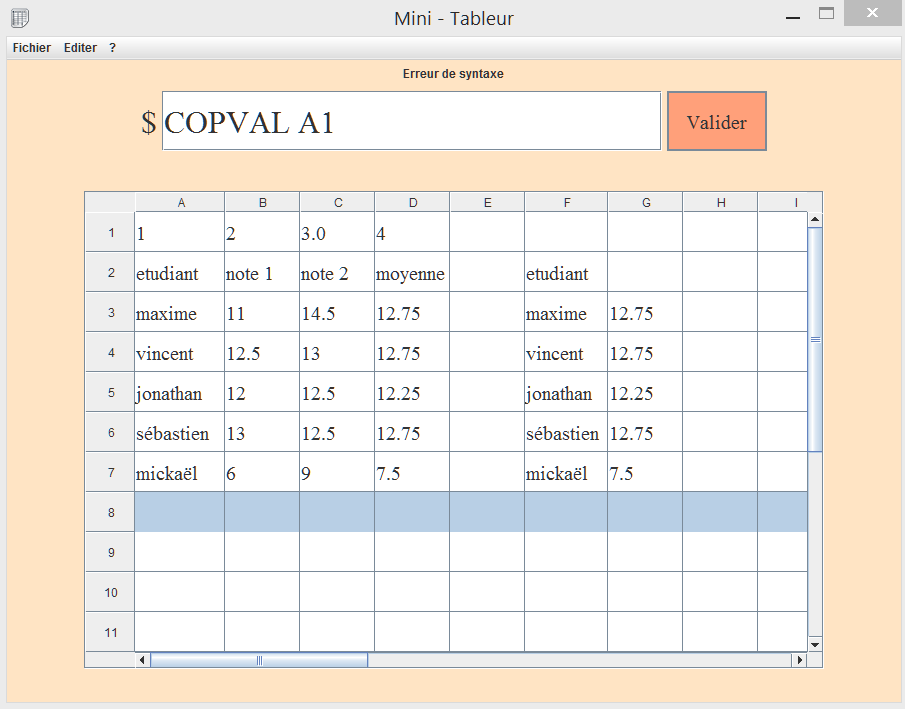
COPIER PLAGE PLAGE : copie la plage source éventuellement tronquée dans la plage destination



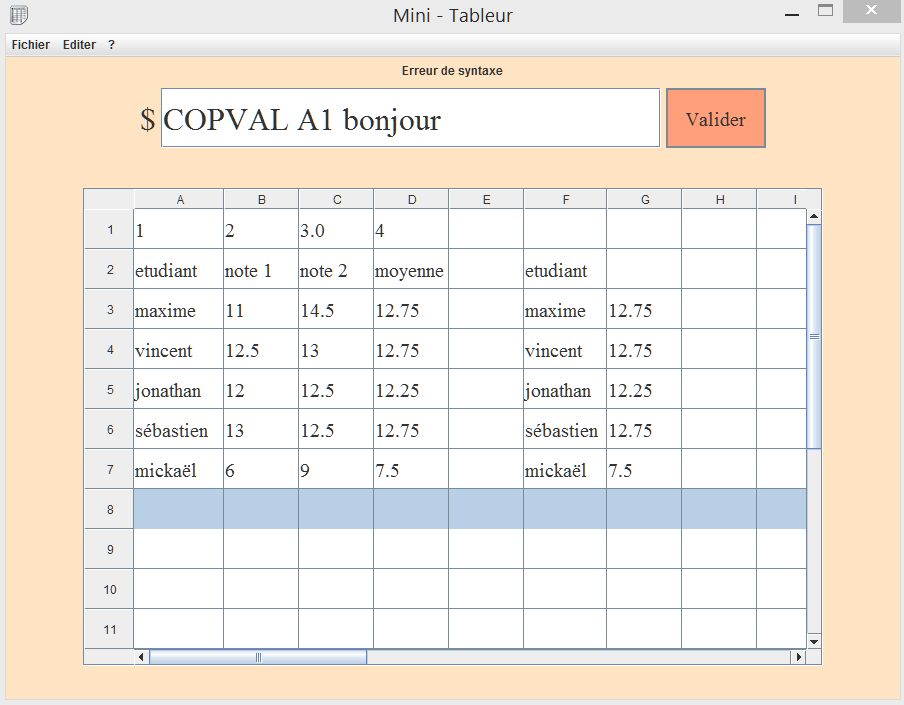
#### - Erreurs :

##### COPVAL

Une erreur survient si un nombre incorrect d’arguments est entré :

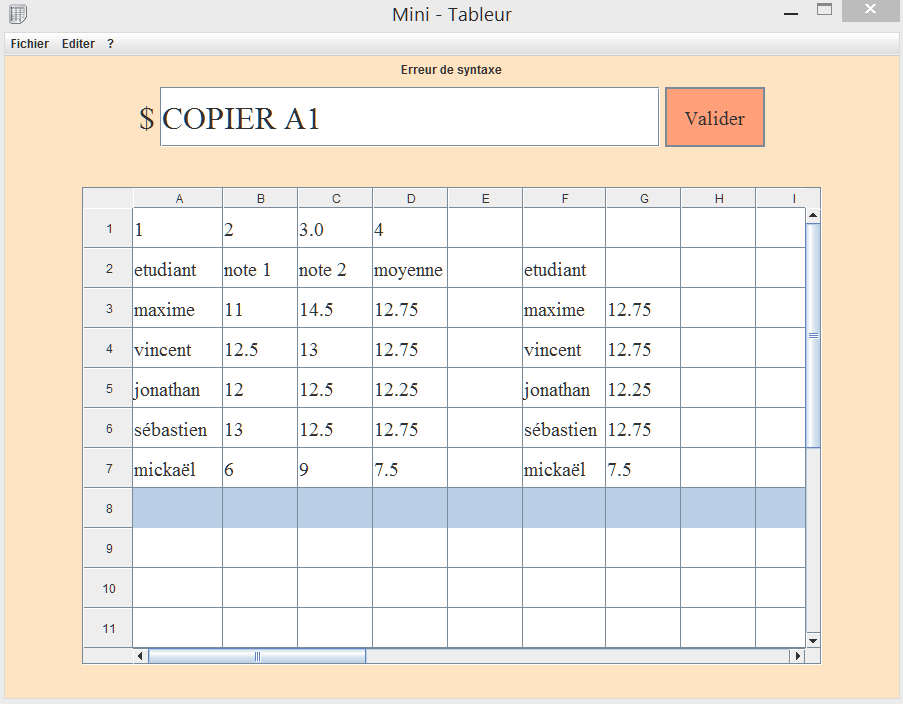
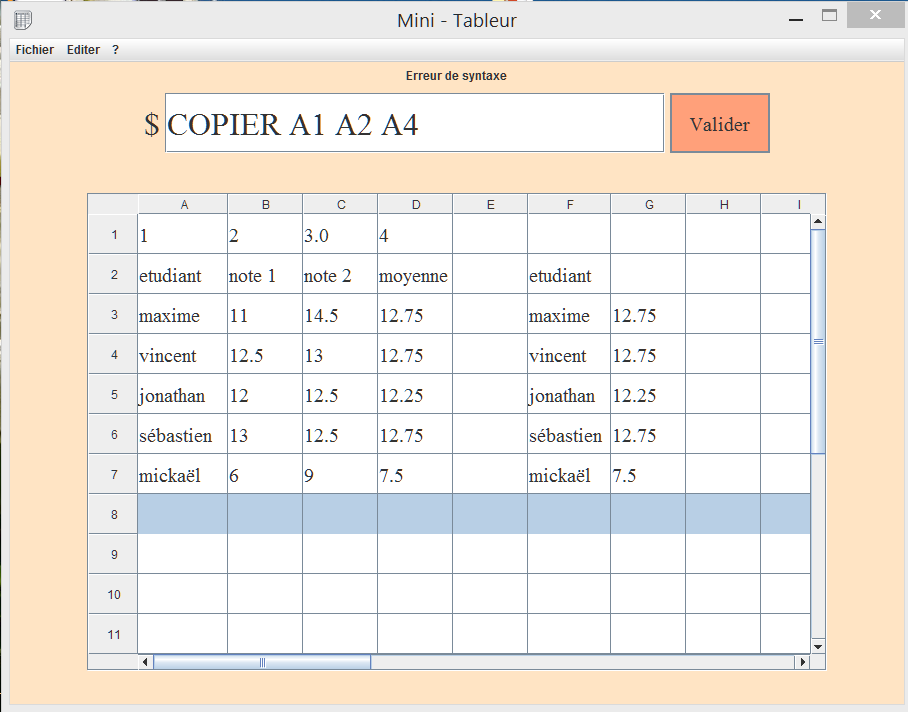


Une erreur survient si un des arguments est incorrect :

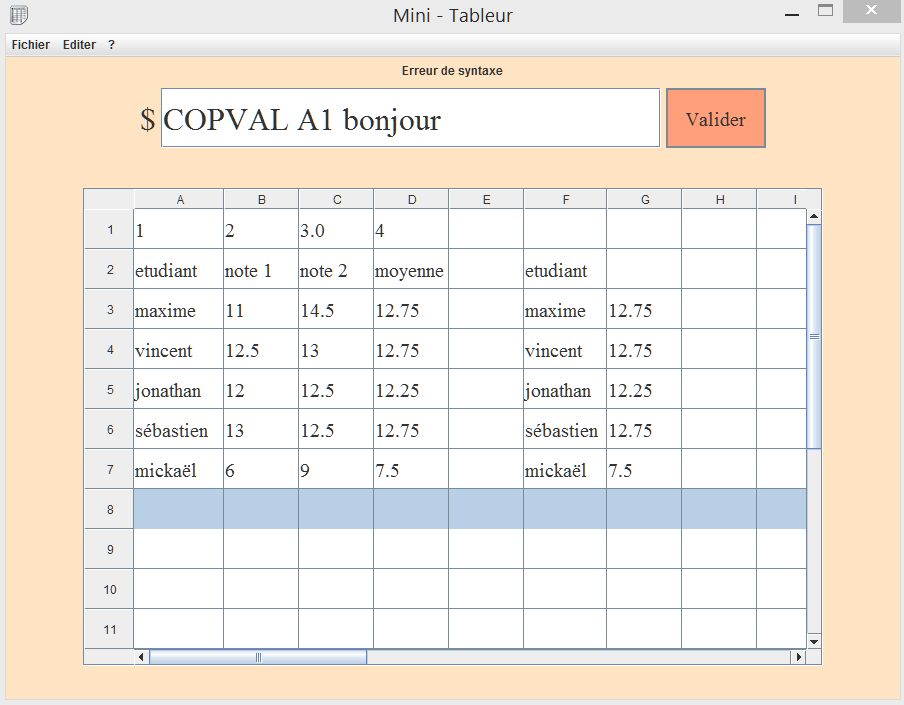


##### COPIER :

Une erreur survient si un nombre incorrect d’arguments est entré :

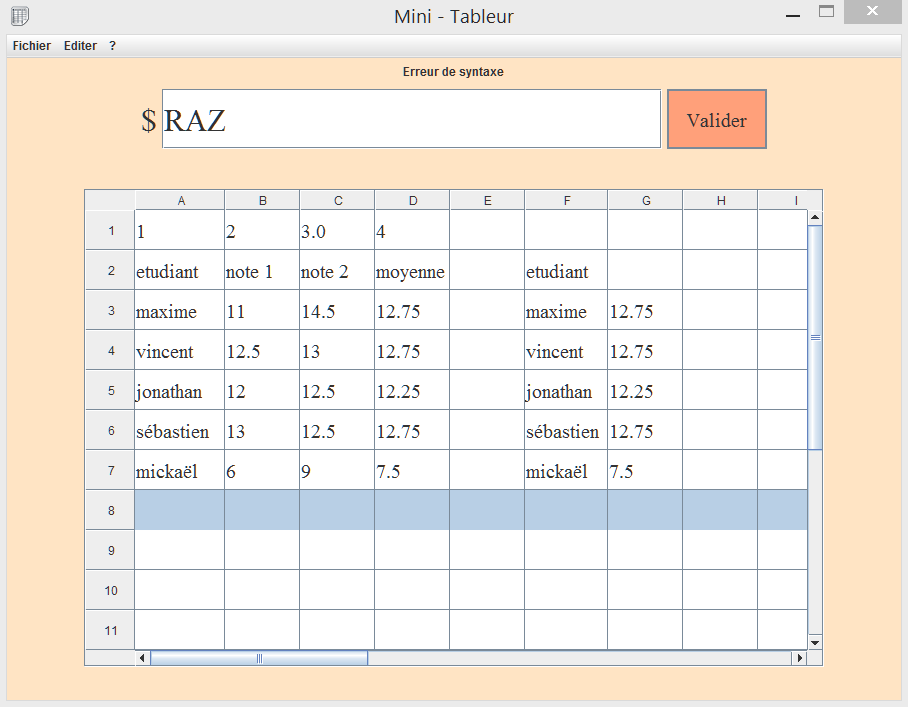
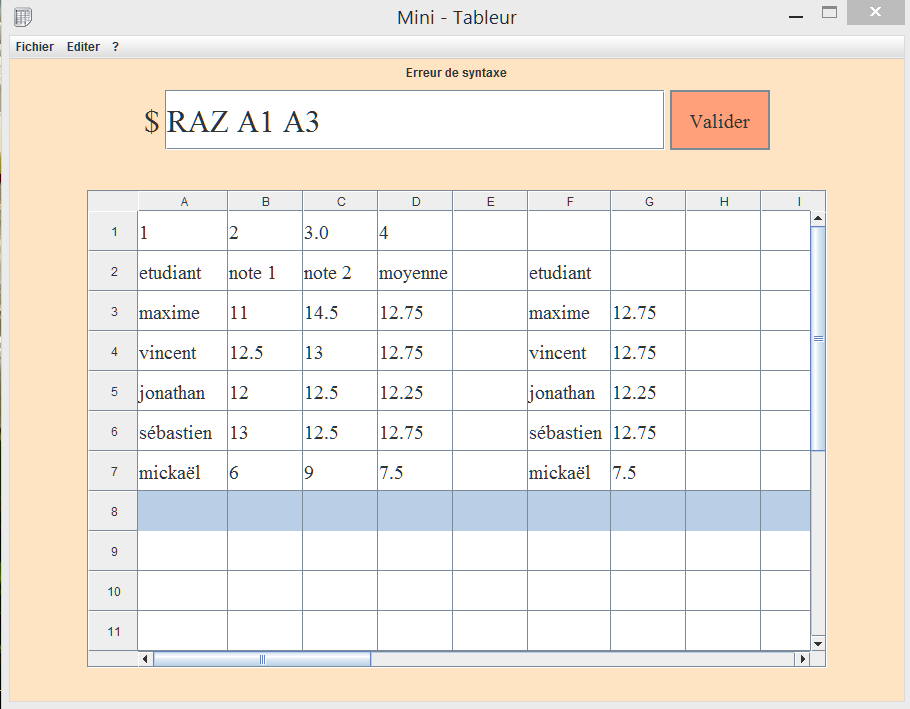


Une erreur survient si un argument est incorrect :

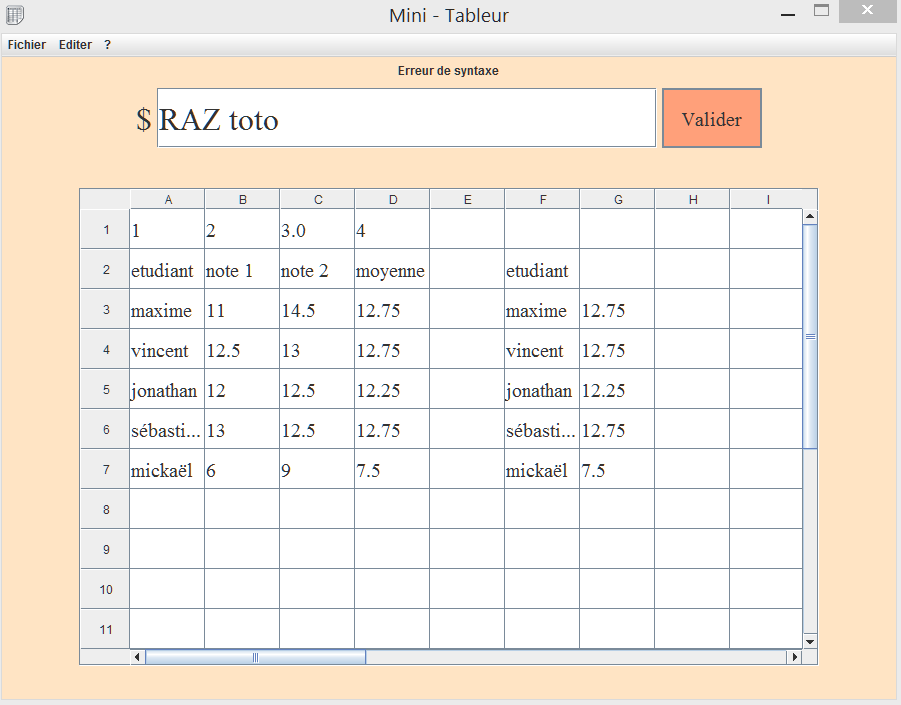


##### RAZ :

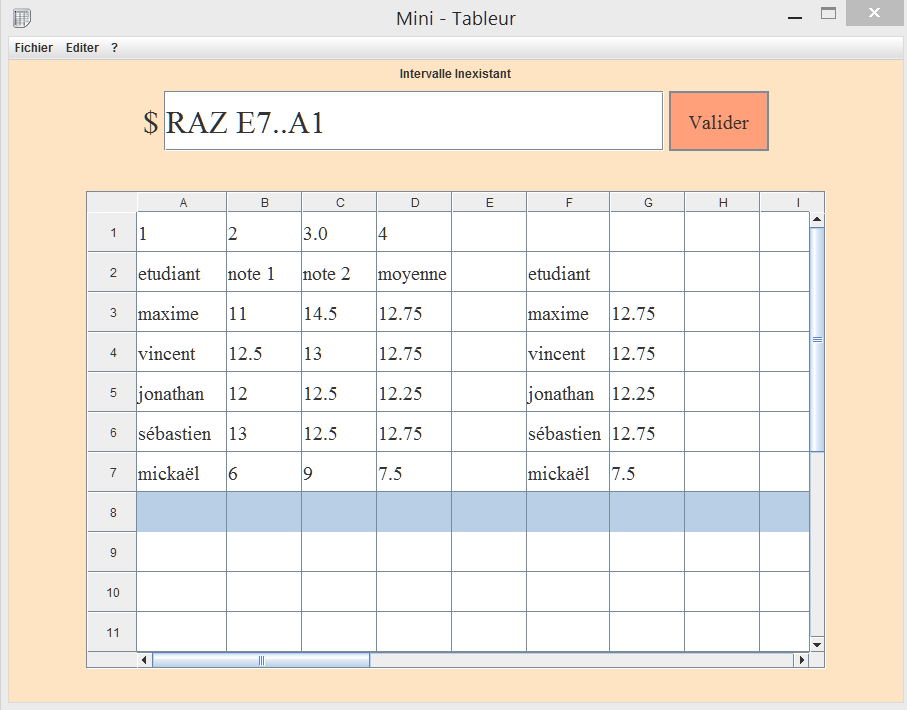
Une erreur survient si un nombre incorrect d’arguments est entré



Une erreur survient si l’argument est incorrect

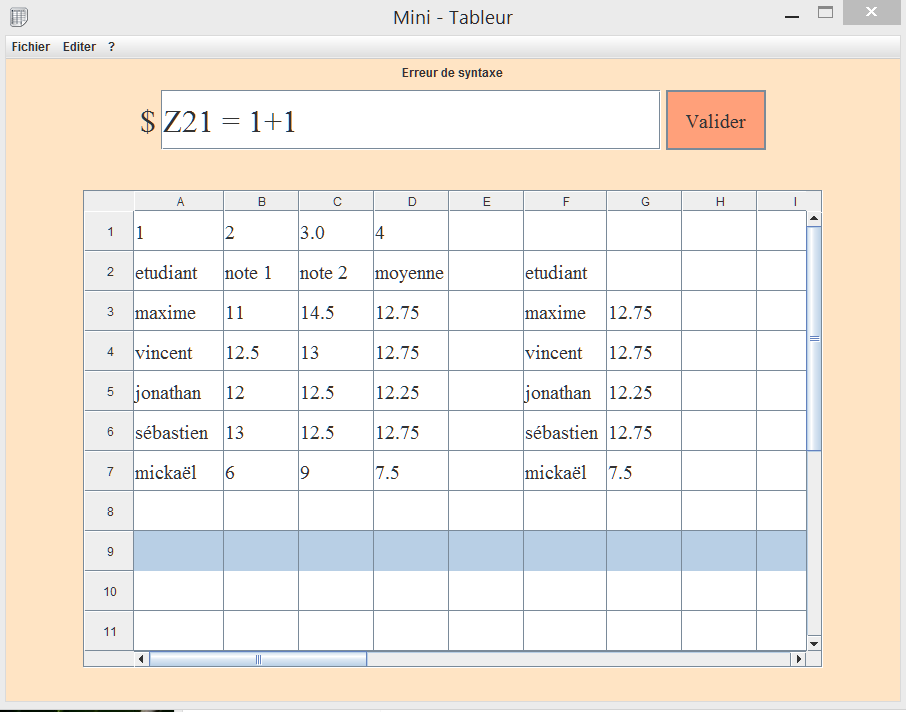


Une erreur survient également si une plage irréelle est entrée en argument



#### Autres erreurs :

Appel d’une case inexistante



# F- Bilan

## 1) Bilan commun

🡪 Satisfaction des besoins

Au terme du projet, la totalité du cahier des charges, définit par la MOA, a été traitée et respectée. Malgré cela, quelques écarts ont été réalisés. En effet, lors des réunions avec la MOE, il a été convenu de modifier ce cahier des charges afin de privilégier le développement de l’application en interface graphique plutôt que sur la console. De plus, nous avons choisi de permettre à l’utilisateur de réaliser des opérations avec un nombre illimité de niveaux d’imbrication de parenthèses. Il en est de même pour le nombre d’opérateurs et d’opérandes. Par ailleurs, d’autres fonctionnalités ont été développées pour offrir à l’utilisateur plus de possibilités. Ainsi, le tableur propose de charger des fichiers ainsi que d’en sauvegarder. Le clic droit est également disponible (sur la calculatrice aussi) afin de permettre des actions d’édition telles que « copier », « couper », « coller ».

Au cours des différents comités de pilotage, le représentant de la MOE a semblé satisfait des avancées réalisées, des prototypes présentés ainsi que des choix entrepris par le groupe.

Le groupe de projet est donc satisfait de ce qu’il a pu réaliser au cours de la réalisation de cette application.

🡪 Bilan de la gestion de projet

**La planification prévisionnelle** n’était finalement pas très éloignée de l’avancement effectif du projet. Les tâches les plus compliquées ont été l’apprentissage de l’utilisation des interfaces graphiques étant donné qu’aucun membre du groupe n’en avait auparavant réalisé dans ce langage, ainsi que les calculs comportant des parenthèses qui nous ont posé beaucoup de soucis.

Malgré cela, le respect des précédences n'a pas toujours été vérifié. En effet la partie concernant la Conception Orientée Objet de l’application, notamment les spécifications et les cas d’utilisation, n’a pas pu être réalisée au début, étant donné que nous n’avions pas encore abordé tous les éléments nécessaires en cours. De ce fait, la réalisation de cette partie a été retardée sans toutefois empêcher l’avancement des autres tâches.

Par ailleurs, **la planification du projet** ne montre pas toutes les retouches et détails apportés aux tâches déjà réalisées. Ces modifications ont souvent découlé de remarques évoquées lors des réunions avec la MOE.

Concernant **la gestion de communication**, nous n’avons pas rencontré de problème pénalisant l’avancement du projet. Le fait d’utiliser des outils de communication, tel que la messagerie instantanée Facebook pour le partage d’idées et d’informations, et des outils de partage, tel que GitHub, a permis de maitriser l’organisation du travail et de gérer la répartition des tâches.

Nous avons tout de même rencontré quelques problèmes. Tout d’abord, nous n’avons pu organiser que très peu de réunions. Enfin, l’utilisation de git nous a donné quelques sueurs froides puisque le changement de JRE au cours du projet a entraîné des conflits dans les classes, nous obligeant à récupérer les fichiers réalisés la veille. Ce problème a donc été source d’une importante perte de temps de travail puisque nous avons perdu la quasi-totalité du travail effectué lors de cette journée.

Au niveau de la gestion humaine, la répartition des tâches s’est faîte presque naturellement, en fonction des capacités de chaque membre du groupe. En effet, les écarts de niveau de programmation imposaient que les tâches les plus complexes soient réalisées par les membres les plus aptes.

Tout au long du projet, la cohésion et l’entente au sein du groupe ont été bonnes. Cette cohésion a été facilitée par le fait que nous nous entendions déjà bien.

Le projet a vraiment été bénéfique pour chacun des membres du groupe puisque tout le monde a énormément progressé en programmation, conception, gestion de projet, IHM, travail en groupe et communication. De plus, nous avons découverts et appris à maitriser de nouveaux outils informatiques.

Nous avons donc pu avoir un premier aperçu de la réalisation d’un projet informatique en entreprise. Cela nous a permis de mieux appréhender certains concepts qui n’étaient pas maitrisés avant la réalisation du projet.

## 2) Bilans personnels

### - Sébastien SANCHEZ

Ce projet a été une très bonne expérience. D'une part d'un point de vues connaissances, puisque j'ai beaucoup appris en matière de programmation évènementielle et de conception d'interface graphiques en Java. J'ai également appris beaucoup sur la gestion d'un projet. Etant chef de projet, il était très important pour moi que l'entente au sein de l'équipe soit optimale, et ce fut le cas. Le travail en groupe a également été abordé avec l'utilisation de Git, qui a été un outil formidable pour permettre à tout le monde de travailler simultanément sur un code, mais qui est très difficile d'utilisation au premier abord.

L'étude de Git et des bibliothèques SWING et AWT sera de plus très bénéfique dans le futur, puisque utiles en entreprise.

L'ambiance au sein du groupe a vraiment été superbe, et de nombreux fous rires ont été au rendez-vous au cours de nos séances de programmation intenses et groupées.

### - Jonathan MIQUEL

Ce projet fût une expérience très enrichissante ajoutant à la programmation dont on avait l'habitude plusieurs dimensions, en effet, de la conception orientée objet jusqu'à la gestion des délais en passant par la répartition des tâches n'avaient jamais eu l'occasion d'être appliqués de manière si concrète. Cela m'a permis de voir la programmation d'un autre angle de vue : la programmation en équipe. C'est un aspect très intéressant qui permet de découvrir la façon dont les autres personnes abordent un problème et ainsi tirer le meilleur de chaque partie. Je garde de ce projet un très bon souvenir, grâce à la bonne entente dans le groupe et la volonté générale de rendre un projet propre.

### - Maxime BAUBE

Ainsi ce projet fut riche en enseignements. En effet, que ce soit en programmation événementielle, avec la découverte et la pratique des bibliothèques AWT/Swing, ou bien en gestion de projet, avec l’utilisation de logiciels tels que Microsoft Project, ou en conception orientée objet avec Visual Paradigm on peut aisément dire que ce projet offre une expérience large et enrichissante au niveau des compétences.

Cependant il ne faut pas oublier le travail de groupe qui s’est déroulé sans accroc tout le long du projet, entre séances de programmation, réunions de gestion de projet ou bien conception tout s’est passé dans une ambiance conviviale.

Néanmoins si je dois mettre un point négatif à ce projet, je citerai la gestion de Git qui malgré son utilité évidente pour le travail de groupe, nous a posé de nombreux soucis internes.

### - Vincent GRANIER

Ce projet a vraiment été une importante source d’évolution et de progression dans les domaines de la programmation, de la conception, de la gestion de projet… . Cela m’a donc permis d’illustrer tous les concepts vus en cours et d’en découvrir de nouveaux.

De plus, la réalisation de ce projet m’a permis d’avoir un premier aperçu de ce qu’est la réalisation d’un projet en entreprise. Par ailleurs, nous avons appris à travailler avec de nouveaux outils informatiques, notamment Project ou encore GitHub. Nous avons donc enrichit notre culture informatique.

Nous avons également progressé dans la gestion humaine. Le travail en équipe et la répartition des tâches nous ont permis de mieux appréhender la réalisation du projet.

Sur un plan personnel, ce projet est donc une réelle réussite puisque j’ai pu combler une partie des lacunes que j’avais en programmation et progresser dans plusieurs domaines de l’informatique.

### - Mickaël PERIES

Au cours de ce projet de quelques mois, j’ai pût apprendre énormément de nouvelles choses. Au niveau informatique grâce à l’utilisation de nouveaux outils comme GIT/Github/Project, mais aussi grâce à l’utilisation de nouveaux outils comme eGit/Metrics. De plus j’ai acquis de nouvelles connaissances comme l’utilisation de nouvelles classes en java (AWT/SWING/io), la création de .exe, etc… De plus j’ai appris à travailler en équipe, ce qui comprend beaucoup de taches comme les réunions, la répartition des tâches, la mise en commun, etc…

J’ai également fortement développé mes capacités d’expression orale, d’écoute, d’assurance et surtout mes capacités rédactionnelles. Ce projet aura été un véritable challenge qui m’a permis de me surpasser pour arriver à respecter les échéances données. De plus je pense que cette simulation de travail en entreprise pourra m’être utile dans le futur. Enfin grâce à une bonne cohésion de groupe nous avons pût exploiter le potentiel de chacun pour obtenir un projet finit et fonctionnel.